

Non appena abbiamo preannunciato sulla rivista che avremmo presentato un microcomputer, sulla nostra redazione hanno cominciato a piovere centinaia di lettere e telefonate che chiedevano chiarimenti in proposito e da queste lettere e telefonate ci siamo ben presto accorti che pochissimi erano coloro che avevano le idee ben chiare in proposito.

Certo se non si sa cos'è una CPU, perché serve una EPROM programmata e a cosa servono le RAM presenti nel circuito, è molto difficile che si riesca un domani ad utilizzare con profitto il microcomputer, infatti ogni volta

che si scrive un programma o anche una sola istruzione, occorre conoscere con esattezza le operazioni che tale programma o tale istruzione comportano all'interno della « macchina », diversamente si riuscirà al massimo a far eseguire al microcomputer un'addizione fra due numeri e finito lì.

Noi invece vogliamo che il nostro microcomputer non rimanga per voi un giocattolo « incomprensibile » che una volta montato, si appoggia su un tavolo o uno scaffale e si lascia quindi in disuso a coprirsi di polvere, bensì vogliamo che tutti riescano a farlo funzionare ed a sfruttarlo al 100% delle sue prestazioni, quindi anche se sappiamo che l'opera che ci attende non è certo delle più facili, cercheremo in parole povere e con semplici analogie di farvi diventare degli esperti conoscitori di questa meravigliosa macchina.

Tanto per iniziare dimentichiamoci per un attimo dell'elettronica e supponiamo che la parola MICROCOMPUTER corrisponda a RISTORANTE CINESE cioè un ristorante all'interno del quale vi sono camerieri, cuochi, direttore di



MICRO

sala, servitori, portiere, cassiera ecc. che non sanno una parola di italiano però sono qui in Italia e debbono servire dei clienti i quali a loro volta non sanno una parola di cinese.

Solo il padrone di questo ristorante è un italiano il quale, considerando che gli riesce molto difficile parlare il cinese, quindi farsi intendere dai suoi dipendenti, ha deciso di assumere un interprete e poiché non ha potuto trovarne uno italiano, ha assunto un francese che sa parlare il cinese.

A tale interprete spetterà ovviamente il compito di parlare con i clienti all'interno del ristorante e poiché questo, essendo francese, non comprende nessuna parola di italiano, il padrone ha escogitato uno stratagemma molto semplice, cioè sui fogli del menù ha scritto le varie voci in italiano e di fianco ha riportato la corrispondente traduzione in francese, per esempio:

VINO ROSSO = VIN ROUGE

BIRRA = BIERE

dopo il caffè si può servire un amaro o se nel conto occorre aggiungere l'IVA, perché in Cina HUA-LIU-FENGH, ministro delle finanze, non l'ha ancora introdotta.

Proprio perché non sa nulla delle nostre abitudini il direttore di sala, pur eseguendo alla lettera qualsiasi istruzione gli venga fornita, si astiene dal prendere decisioni o iniziative che non gli siano state espressamente indicate onde evitare di fare delle « gaffe ».

Il padrone del locale è al corrente di questo e per evitare brutte sorprese gli ha preparato, con l'aiuto del traduttore francese, un block-notes su cui sono scritti (in cinese) uno per uno tutti i movimenti e le operazioni che esso deve compiere dal momento dell'apertura del locale fino alla chiusura.

A questo punto facciamo una breve sosta di riepilogo e vediamo di paragonare i personaggi di questa nostra « storiella » con gli elementi del microcomputer in modo da poter meglio inquadrare il problema.

— Il **MICROCOMPUTER** è il ristorante cinese nel suo

COMPUTER Z80

Dopo avervi presentato, sul n. 68, i primi tre telai del nostro microcomputer, cercheremo oggi di spiegarvi le funzioni principali che esplicano i vari stati di questo circuito, nonché di farvi prendere confidenza con la programmazione, una parola questa per molti assolutamente nuova.

In fondo al menù ha poi scritto una nota in cui prega i clienti di apporre una crocetta di fianco alla voce prescelta in modo tale che i camerieri li possano servire.

In pratica il cliente sceglie per esempio il vino rosso e fa una crocetta a matita di fianco a questa voce.

L'interprete francese legge « vin-rouge » e traduce questa frase in cinese poi va a scriverla su una lavagna di servizio che il padrone ha disposto all'interno del locale.

Una volta risolto il problema della scelta dei vari piatti da parte del cliente, per il gestore del locale ne resta un secondo da risolvere e precisamente quello di istruire il **direttore di sala**, che come abbiamo detto è un cinese e non capisce una sola parola di italiano, in modo che questo possa dirigere nel migliore dei modi la vita del locale.

Inutile aggiungere che questo direttore non sa nulla delle nostre abitudini, cioè non sa se la pasta siamo abituati a mangiarla con la forchetta o con i bastoncini di legno, se a tavola beviamo il vino oppure il the, se la minestra va servita in un piatto oppure in una tazza, se

complesso.

— La **C.P.U.** è il direttore di sala cinese

— Le **UNITÀ di INGRESSO/USCITA**, cioè gli integrati che fanno da corredo alla CPU e servono per l'ingresso e l'uscita dei dati da o verso la memoria e le periferiche, sono tutto il personale cinese del ristorante, vale a dire camerieri, cuochi, cassiera ecc.

— Il **CODICE BINARIO**, cioè il linguaggio macchina, è la lingua parlata dai camerieri e dal direttore di sala, cioè il cinese.

— I **TECNICI della NUOVA ELETTRONICA** che hanno costruito il microcomputer sono in pratica il padrone del ristorante.

— La **EPROM** già programmata è il « taccuino » su cui i tecnici hanno scritto in binario, cioè in cinese, le istruzioni per il direttore di sala, cioè per la CPU.

— La **TASTIERA ESADECIMALE** è l'interprete francese assunto dal proprietario per poter parlare con i suoi dipendenti e soprattutto per poter parlare con i clienti del



locale, cioè con **VOI LETTORI**.

— Le **TABELLE** che noi vi forniremo con tutte le istruzioni e di fianco il relativo codice esadecimale, sono invece il « menù » scritto in due lingue.

— La **MEMORIA RAM** è infine la lavagna presente all'interno del ristorante, pronta a ricevere gli ordini che voi farete ai camerieri.

Vediamo ora cosa succede all'apertura di questo locale tenendo presente che le medesime cose avvengono più o meno all'interno del microcomputer quando si fornisce tensione.

Precisiamo che nessuno dei dipendenti, cioè camerieri, cuochi, ecc. muove un solo passo senza che glieli ordini il **direttore di sala**, cioè la CPU, in quanto il proprietario ha instaurato nel locale una disciplina ferrea e non ammette che nessuno disobbedisca.

La prima cosa che fa il direttore di sala, cioè la CPU, è guardare nel suo taccuino (la EPROM) e leggere la prima istruzione in cinese (cioè in binario):

« Apparecchiare tutti i tavoli con tovaglia, tovaglioli, forchetta, coltello, bicchieri, saliera e contenitore per il pane ».

Il direttore chiama quindi i camerieri e ordina loro di apparecchiare tutti i tavoli in questo modo, poi va a vedere nel suo taccuino e legge la seconda istruzione:

« Aprire la porta d'ingresso e mettersi tutti in attesa del primo cliente ».

Esso ordina pertanto ad un servitore di aprire la porta, poi ordina ai camerieri ed all'interprete di tenersi pronti.

A questo punto entra il primo cliente nel locale e il direttore di sala si affretta a leggere sul taccuino (cioè sulla EPROM) la successiva istruzione:

« Mandare un cameriere al tavolo con la caraffa del vino poi mandare l'interprete con il menù e chiedere al cliente cosa vuole mangiare ».

La CPU, cioè il direttore di sala, ordina pertanto ad un cameriere, cioè ad uno degli integrati che gli stanno attorno, di eseguire la prima istruzione, cioè di portare il vino al cliente poi ordina all'interprete di avvicinarsi e chiedergli cosa desidera da mangiare.

Il cliente, cioè l'operatore esterno, guarda il **menù** (cioè la tabella in cui sono riportati i codici esadecimali relativi alle varie istruzioni) e sceglie un piatto qualsiasi, per esempio « affettato misto » facendo un crocetta di fianco (in pratica l'operatore imposterà sulla tastiera esadecimale il codice relativo all'istruzione prescelta).

L'interprete legge di fianco a « affettato misto » la traduzione francese di questa parola ed a sua volta la traduce in cinese, per il direttore di sala il quale incarica un cameriere di scriverla sulla lavagna, cioè sulla RAM, poi guarda di nuovo nel suo taccuino e legge la successiva istruzione:

« Chiedere al cliente cosa desidera come minestra »

Esso ordina pertanto al traduttore di chiedere al cliente cosa vuole come minestra ed ammesso che il cliente scelga per esempio un **piatto di riso** facendo una crocetta di fianco a questa voce, il traduttore leggerà prima il corrispondente francese di questa parola, poi la tradurrà in cinese per il direttore di sala e questi a sua volta incaricherà un cameriere di scriverla sulla lavagna in cinese, cioè in binario.

In pratica questo dialogo fra direttore di sala, interprete e cliente si protrarrà fino a quando il cliente dirà STOP, cioè « non voglio altro » ed a questo punto sulla lavagna, cioè sulla memoria RAM, vi saranno scritte tante righe in cinese (cioè in codice binario) quanti sono i piatti che il cliente ha richiesto.

In altre parole il cliente, cioè noi che lavoriamo dall'esterno, abbiamo consegnato al microcomputer (cioè al ristorante) tutte le istruzioni e i dati per poterci servire ed ora il personale si appresta a farlo nel migliore dei modi.

UNITÀ DI INGRESSO-USCITA



SQUATTERO

CUOCO

Fig. 1 Un microcomputer può essere paragonato ad un ristorante « cinese » che per funzionare ha bisogno di un interprete (la tastiera esadecimale) che traduca automaticamente in cinese ciò che noi ordiniamo in italiano, una lavagna (memoria RAM) su cui scrivere in cinese i vari ordini e un direttore (la CPU) che leggendo sulla lavagna ed aiutandosi col suo taccuino personale (la EPROM), coordina il lavoro di tutti gli altri dipendenti in modo da poter offrire al cliente quanto esso richiede.

Se potessimo leggere sulla lavagna il nostro ordine, troveremmo riportato su tante righe successive quanto segue:

00010110 = affettato misto
 10010001 = riso
 11000111 = nidi di rondine
 00100000 = dolce
 01000001 = caffè
 11111111 = stop

Quando il cliente ha terminato il suo ordine, il direttore riprende il suo notes e comincia a leggere e ad eseguire le istruzioni successive.

1) **Leggi la prima riga sulla lavagna**

Il direttore guarda la lavagna e legge « affettato misto ».

2) **Fai preparare questo piatto al cuoco.**

Il direttore chiama un cuoco e gli fa preparare un piatto di affettato misto.

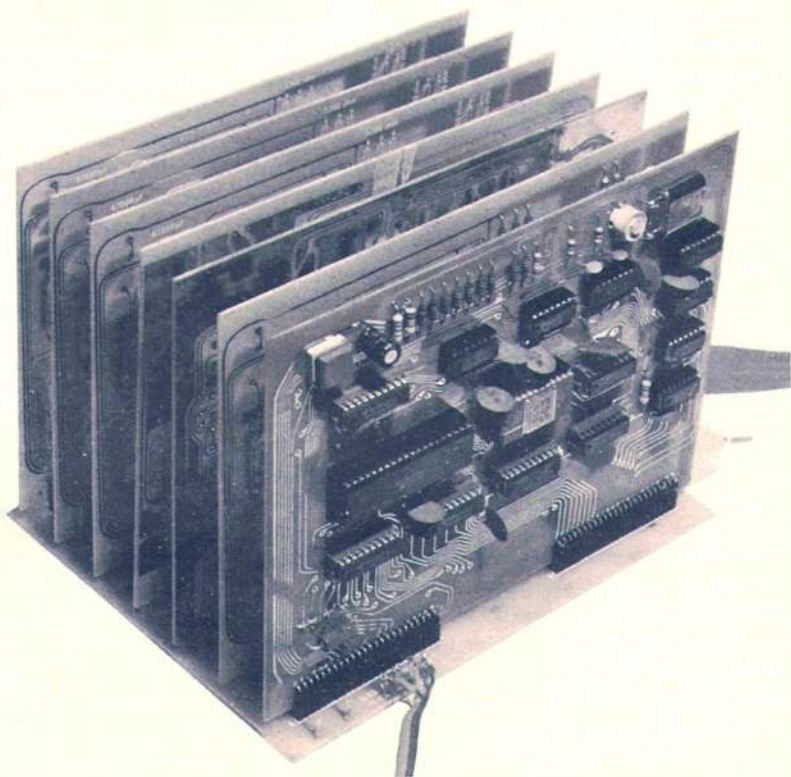
3) **Chiama un cameriere e fai portare il piatto al cliente.**

Il direttore chiama un cameriere e gli fa portare al cliente il piatto di affettato misto.

4) **Quando il cliente ha finito l'antipasto chiama lo sguattero e fagli portare via il piatto sporco, poi leggi la seconda riga sulla lavagna e fai preparare al cuoco questa portata.**

Il direttore osserva il cliente e quando questo ha finito l'antipasto manda lo sguattero a togliere il piatto, poi legge sulla lavagna « riso » e incarica il cuoco di prepararlo.

Attualmente sul nostro « BUS » (piastra di supporto) sono inserite due sole schede, cioè la scheda CPU e l'interfaccia tastiera, ma in seguito, quando vi avremo presentato l'interfaccia per il registratore a cassette, quella per il video e la tastiera alfanumerica, le schede di espansione della memoria, l'interfaccia stampante ecc., il BUS stesso si riempirà totalmente come vedesi in questa foto.



5) Fai servire al cliente la minestra

Il direttore chiama un cameriere e lo manda a portare il piatto di riso sul tavolo del cliente.

6) Quando il cliente ha finito di mangiare la minestra chiama lo sguattero e fagli portare via il piatto sporco, poi leggi la terza riga sulla lavagna e fai preparare al cuoco questa portata.

Il direttore osserva il cliente e quando questo ha finito la minestra, manda lo sguattero a togliere il piatto, poi incarica il cuoco di preparare i nidi di rondine.

7) Fai servire al cliente il secondo.

Il direttore incarica un cameriere di portare i « nidi di rondine » sul tavolo del cliente.

8) Quando il cliente ha finito di mangiare il secondo chiama lo sguattero e fagli portare via il piatto sporco e le posate poi leggi sulla lavagna la quarta riga e guarda se ha ordinato il dolce.

Il direttore attende ancora che il cliente abbia finito poi incarica lo sguattero di portare via piatto e posate, quindi vedendo che il cliente ha richiesto il dolce lo fa preparare dal cuoco.

9) Fai portare al cliente.

Il direttore incarica un cameriere di portare il dolce al cliente.

10) Guarda sulla lavagna se il cliente ha richiesto il caffè ed in caso affermativo aspetta che abbia finito di mangiare, fagli togliere il piatto poi manda qualcuno a prepararlo.

Il direttore attende il momento giusto poi incarica un cameriere di preparare il caffè e portarlo al cliente.

11) Leggi la sesta riga sulla lavagna e se trovi scritto STOP prepara il conto includendo tutto ciò che è scritto sulla lavagna.

Il direttore trova effettivamente lo STOP, quindi legge sulla lavagna tutte le porzioni consumate dal cliente, controlla sul suo notes personale il relativo prezzo ed esegue l'addizione.

12) Aggiungi il 14% di IVA.

Il direttore esegue il calcolo e addiziona al totale il 14% di IVA.

13) Consegna il conto all'interprete e fallo portare al cliente.

Il direttore chiama l'interprete, gli consegna il conto e lo manda dal cliente per farsi pagare.

14) Fai sprecchiare la tavola e falla quindi preparare per un nuovo cliente.

Il direttore manda lo sguattero a sprecchiare la tavola, quindi chiama un cameriere e lo manda ad apparecchiare di nuovo con bicchieri e posate pulite.

A questo punto nel ristorante ci si mette in attesa di un nuovo cliente che si siede a tavola e ordini un pranzo ed anche per questo tutto l'ordine verrà trascritto sulla lavagna nelle righe successive.

Poiché la RAM utilizzata, cioè la lavagna, risulta da 1 K, vale a dire dispone di 1024 righe, una volta che si saranno

riempite tutte queste righe si ricomincerà daccapo, cioè si cancellerà la prima riga della lavagna e su questa si scriverà il nuovo piatto richiesto, poi si cancellerà la seconda e così di seguito.

Tale modo di procedere può essere però molto pericoloso, infatti se si cancellano alcune righe della lavagna prima ancora che il direttore abbia potuto fare il conto al relativo cliente, questo non sarà più in grado di farlo perché non potrà ricordarsi cosa aveva ordinato il cliente, oppure gli farà pagare ciò che esso non ha consumato e che ha richiesto invece il cliente giunto per ultimo.

Quindi se i clienti da servire sono tanti che esiste la possibilità di saturare la lavagna, piuttosto che correre di questi rischi, meglio aggiungere subito in parallelo alla prima una seconda oppure una terza lavagna, ottenendo così un totale di 2.000 oppure 3.000 righe di memoria.

Questo dovrebbe farvi comprendere perché nel nostro microcomputer abbiamo previsto l'espansione della memoria RAM fino ad un massimo di **64 K**, cioè **64.000 righe**.

A questo punto riteniamo che abbiate compreso quale funzione svolgono all'interno del microcomputer la CPU, la EPROM, la RAM e la TASTIERA ESADECIMALE, però se vi restano ancora dei dubbi cercheremo di dissiparli con ulteriori esempi.

Prendiamo ad esempio la EPROM: come avete potuto apprendere questa è programmata per il lavoro che deve svolgere, cioè gli sono state scritte delle istruzioni che per noi sembrerebbero ovvie, mentre per il direttore di sala (cioè la CPU) e per i camerieri (cioè gli integrati che gli stanno attorno), non lo sono affatto.

Per esempio quando il cliente ha ordinato il dolce non ha detto di cambiargli le posate ed il piatto, però il direttore leggendo nel suo notes vi ha trovato scritto che il dolce va servito in un piatto a parte con il relativo cucchiaino, quindi ha fatto eseguire ai camerieri questa istruzione.

Quando il cliente ha chiesto il caffè non ha specificato che lo voleva nella tazzina perché questo per lui è ovvio, però il direttore di sala che magari in Cina era abituato a servire il caffè entro una ciotola di legno, ha dovuto consultare il suo notes per eseguire correttamente questa operazione.

Quindi la EPROM già programmata è fondamentale nella vita del microcomputer (cioè per il ristorante) in quanto contiene tutte le istruzioni basilari per consentire al direttore di sala (cioè alla CPU) di adempiere in modo corretto alle richieste del cliente il quale, essendo italiano e non cinese, è abituato a ragionare ed agire in maniera totalmente diversa da lui.

A questo punto però viene spontaneo chiedersi: « va bene, noi abbiamo scritto sul notes del direttore tutte le operazioni che esso deve compiere per servire il cliente, però ammesso che il cliente voglia ordinare qualcosa che non è sul menù, come fa il direttore a risolvere questo problema che non è previsto nel suo promemoria? »

Ebbene anche questo problema nel microcomputer è

risolvibile purché il cliente si adegui alle « regole » del locale.

Supponiamo per esempio che un qualsiasi cliente entri nel ristorante e come minestra desideri per esempio gli « spaghetti alla carbonara » che non esistono nel menù.

Ovviamente, non trovando questa voce, non può fare una crocetta su nessuna casellina del menù stesso e neppure può pensare di scrivere « spaghetti alla carbonara » perché l'interprete francese non saprebbe cosa sono quindi non potrebbe tradurre questa parola in cinese e tornerebbe a chiedergli cosa vuole.

Il proprietario del locale ha però pensato anche a questa evenienza e in fondo al menù ha aggiunto una lista di tutte le possibili parole utilizzate in un ristorante, cioè spaghetti, maccheroni, burro, formaggio, forchetta, coltello, bicchiere, mescolare, cuocere ecc. con accanto la relativa traduzione in francese (cioè in esadecimale) quindi il cliente può servirsi eventualmente di queste parole per spiegare al direttore cosa vuole esattamente che gli si porti.

Ad esempio, per ottenere gli spaghetti alla carbonara dovrà scrivere:

- 1) Prendere pancetta affumicata e tagliare a cubetti
- 2) Mettere in un tegame a rosolare
- 3) Cuocere spaghetti in pentola poi scolare
- 4) Sbattere un uovo
- 5) Aggiungere formaggio, sale e pepe
- 6) Versare su spaghetti
- 7) Aggiungere pancetta e mescolare
- 8) Versare in un piatto poi servire.

In altre parole tutte quelle istruzioni che il direttore non ha scritto nel suo notes personale debbono essere fornite una per una all'interprete il quale provvederà a farle scrivere sulla lavagna dopodiché il direttore, leggendo queste istruzioni, potrà far preparare dal cuoco ciò che il cliente desidera.

Inutile aggiungere che se il cliente si dimentica di scrivere anche una sola di queste istruzioni, per esempio quella di versare la pancetta sugli spaghetti e mescolare, il direttore non potrà inventarsi questa istruzione in quanto, per paura di sbagliare, esso si attiene rigorosamente a ciò che gli è stato ordinato, quindi il piatto di spaghetti verrà servito senza la pancetta e con l'uovo tutto raggrumato da una parte perché non è stato mescolato.

A questo punto qualcuno potrebbe dirci che a lui non interessa costruire un microcomputer per dirigere un ristorante, bensì gli interesserebbe molto di più un microcomputer da utilizzarsi come magazziniere nella propria cartoleria, quindi ci chiederà come deve comportarsi, cioè se deve cambiare la CPU, se gli occorre una tastiera diversa e così di seguito.

Ebbene le operazioni da compiere per trasformare il nostro direttore di sala in un perfetto magazziniere sono molto semplici infatti per ottenere questo è sufficiente cancellare sul suo NOTES le istruzioni che servivano per

gestire il ristorante e sostituirle con altre istruzioni che gli permettano di sbrigare tutte le faccende di un magazzino.

Ovviamente tutti gli elementi che formavano il ristorante, resteranno inalterati anche nel magazzino, cioè avremo la solita lavagna su cui scrivere gli ordini, avremo sempre l'interprete francese, i camerieri si trasformeranno in commessi ed i cuochi in confezionatori di pacchi e tutti continueranno a parlare e a comprendere solo ed esclusivamente la lingua cinese (cioè il binario).

Anche il cliente dovrà adeguarsi a questa nuova circostanza, cioè non potrà entrare nel magazzino e chiedere per esempio un piatto di lasagne perché non essendo prevista una domanda di questo genere nel notes del magazziniere, nessuno saprebbe dargli una risposta, però se chiedesse per esempio un quaderno a righe facendo una crocetta sull'apposito catalogo, il traduttore convertirebbe subito questa frase in cinese e la farebbe scrivere sulla lavagna, poi il magazziniere incaricherebbe un commesso di andare a prendere il quaderno e di portarlo sul banco al cliente, proprio come avveniva nel ristorante.

Supponiamo ora che a qualcuno interessi avere a disposizione un « direttore » (cioè una CPU) così versatile da poter adibire ai seguenti impieghi:

- 1) dirigere un ristorante
- 2) dirigere un magazzino
- 3) dirigere un ufficio
- 4) dirigere un albergo
- 5) dirigere il traffico in una stazione
- 6) ecc. ecc.;

Voi senz'altro risponderete: « è semplicissimo, basta prendere tante EPROM, scrivere su ognuna di esse che cosa deve fare la CPU nelle varie applicazioni, poi a seconda delle esigenze impiegare la EPROM n. 1, la n. 2, la n. 3 ecc. ».

Ebbene questo è senz'altro fattibile però noi possiamo anche fare una seconda cosa e cioè scrivere sul notes del nostro direttore tutte le istruzioni che servono per il ristorante, numerandole una per una, poi tutte le istruzioni che servono per il magazzino, tutte quelle per l'ufficio, tutte quelle per l'albergo ecc. quindi all'inizio del taccuino stesso scrivere un'istruzione di questo genere:

« Prima di fare qualsiasi cosa chiedi all'interprete cosa devi dirigere; se ti risponde « un ristorante » prendi in considerazione solo le istruzioni del primo gruppo; se ti risponde « un magazzino » considera solo quelle del secondo gruppo ecc. ecc. »

In tal caso il direttore, non appena accenderemo il microcomputer, ci chiederà tramite l'interprete:

« Cosa debbo dirigere? »

Noi ovviamente gli risponderemo impostando sulla tastiera il codice esadecimale che corrisponde a ristorante-magazzino-ufficio ecc. a seconda delle nostre esigenze e da questo punto in poi le altre istruzioni riportate sul notes saranno per lui come inesistenti in quanto esso prenderà in considerazione solo ed esclusivamente quelle inerenti

l'applicazione che gli abbiamo richiesto.

Precisiamo che se tutte queste istruzioni non potessero essere contenute in una sola EPROM (la quale ha una capienza massima di 1.024 righe), si potrà sempre collegare in parallelo a quella già esistente una seconda oppure una terza EPROM che troveranno alloggio sull'apposita scheda nel bus di espansione.

Se poi le istruzioni da fornire a questo direttore fossero tante da richiedere un numero eccessivo di EPROM, cioè un taccuino troppo ingombrante, per risolvere il problema potremo adottare anche una seconda soluzione, cioè fornire al direttore un mangianastri con tante cassette su ognuna delle quali sono memorizzate le istruzioni relative ad una diversa applicazione, per esempio sulla cassetta A vi saranno tutte le istruzioni necessarie per dirigere un ristorante, sulla cassetta B tutte quelle per dirigere un magazzino, sulla cassetta C tutte quelle per dirigere un ufficio e così di seguito.

Precisiamo che il direttore non metterà mai in funzione di sua spontanea volontà questo mangianastri per leggere le istruzioni da noi fornite, bensì guarderà sempre il suo notes personale, cioè la EPROM, quindi su tale EPROM

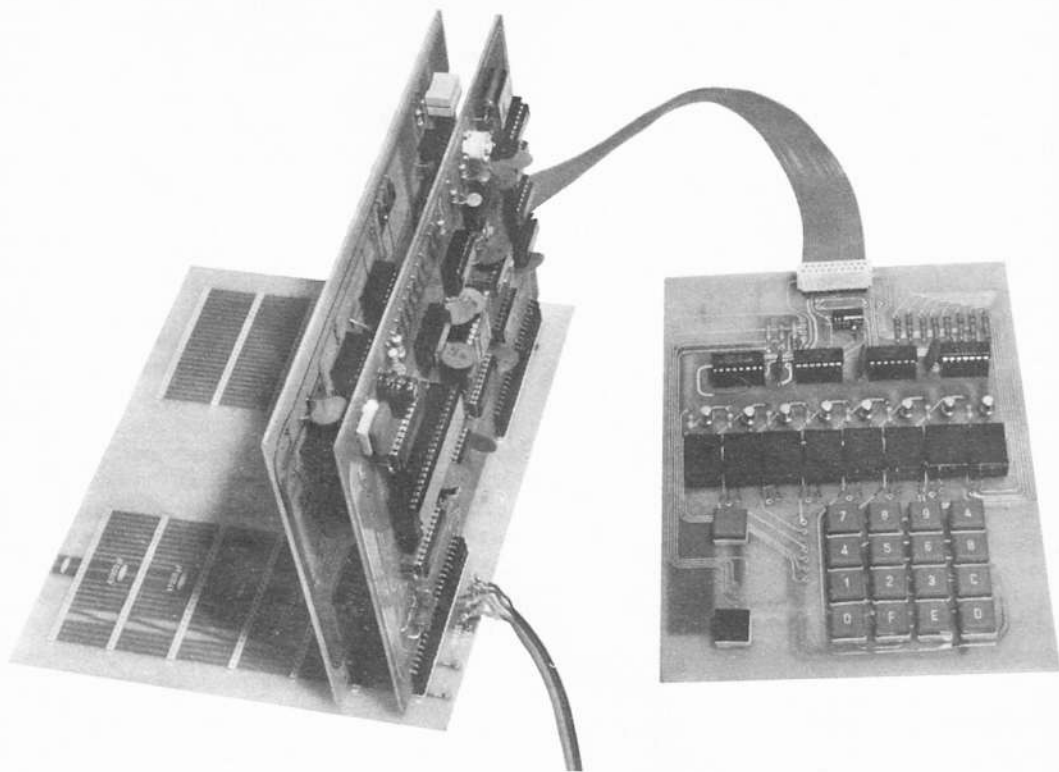
noi dovremo innanzitutto porre un'istruzione di questo genere:

« Leggi sul nastro il programma di lavoro che ti abbiamo assegnato e trascrivilo sulla lavagna, poi esegui una istruzione per istruzione ».

Ovviamente il nastro dovremo fornirglielo noi ogni volta e solo quando lo avremo inserito all'interno del mangiacassette, il direttore potrà andare a leggerlo ed iniziare così ad eseguire le istruzioni in esso contenute.

Inutile aggiungere che se ci sbagliassimo ad inserire il nastro, cioè inserissimo per esempio quello che contiene le istruzioni per gestire il ristorante, mentre ci necessita quello per gestire un magazzino, il direttore lo leggerà egualmente e trascriverà queste istruzioni sulla lavagna però quando cercherà di eseguirle e non troverà ciò che gli serve, per esempio si troverà dei piatti al posto delle matite, oppure delle bottiglie al posto dei quaderni, non saprà più cosa fare e si bloccherà. A questo punto per farlo ripartire noi dovremo inserire nel mangianastri la cassetta giusta, fargliela leggere di nuovo, poi dirgli di eseguire queste istruzioni.

Il sistema di impiegare un nastro magnetico è molto



Nella foto, come si presenta attualmente il nostro microcomputer con le due schede disponibili innestate sul BUS e la relativa tastiera esadecimale. I fili che si vedono in basso sono quelli relativi alle alimentazioni che all'atto dell'accensione debbono assolutamente giungere secondo l'ordine richiesto (in caso contrario si può bruciare la EPROM); proprio per tale motivo, su questo stesso numero, troverete un apposito alimentatore siglato LX380.

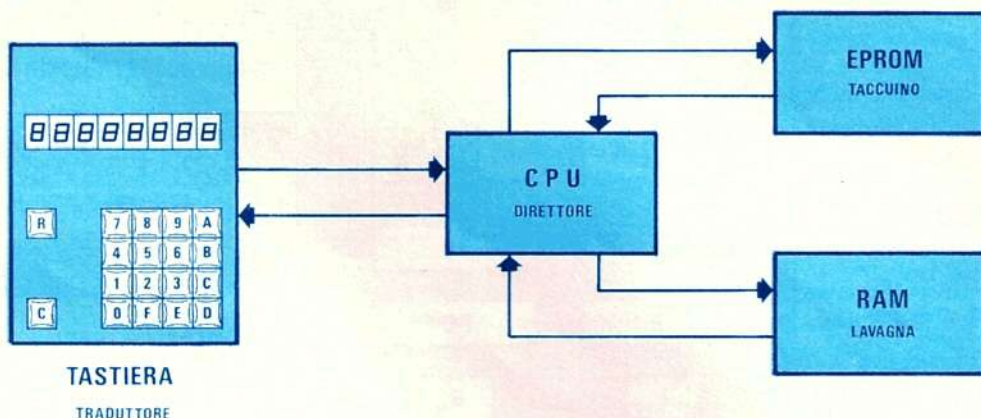


Fig. 2 Nel nostro microcomputer la tastiera esadecimale svolge la funzione di interprete per trascrivere in cinese sulla lavagna (memoria RAM) i nostri ordini. La CPU (cioè il direttore di sala), una volta terminato l'ordine, provvederà quindi a leggere riga per riga quanto risulta scritto sulla RAM ed aiutandosi col suo notes (EPROM) eseguirà tutte le istruzioni memorizzate. Se gli mancasse la EPROM, cioè il suo taccuino personale, la CPU non saprebbe cosa fare e tutto il « ristorante » rimarrebbe paralizzato nessuno essendo in grado di prendere decisioni autonome.

comodo in quando non solo potremo scrivere su tale nastro qualsiasi programma o istruzione che ci necessita, ma avremo anche il vantaggio di poterlo cancellare, modificare, correggere, perfezionare secondo le nostre esigenze con estrema facilità, cosa questa che non potremmo fare su una EPROM che ci viene consegnata già programmata, infatti è vero che è possibile cancellare e modificare anche questa ma l'operazione è sempre oltremodo difficoltosa.

Nella EPROM inseriremo invece i programmi comuni a qualsiasi lavoro, per esempio tutte le istruzioni necessarie al direttore per prendere gli ordini dal cliente, tutte le istruzioni necessarie per fare il conto alla fine, per sommare l'IVA ecc. ecc. cioè tutte quelle istruzioni che servono sia nel ristorante, che nel magazzino oppure nell'albergo.

Il menù, oppure il listino prezzi, dato che varia da applicazione ad applicazione, lo scriveremo infine sempre sul nastro in modo tale da utilizzare ogni volta solo ed esclusivamente quello che fa al caso nostro.

Prima di concludere ricordiamo che un nastro ci sarà utilissimo come « traduttore » quando in sostituzione della tastiera esadecimale impiegheremo una tastiera alfanumerica simile a quella di una macchina da scrivere.

Infatti la CPU, dietro nostro comando, ricopierà nella memoria RAM, cioè sulla lavagna, il contenuto di questo nastro e quando noi scriveremo sulla tastiera: **STAMPA - ADDIZIONE - SOTTRAI** ecc. andrà a cercarsi sulla lavagna tale parola e di seguito troverà la corrispondente traduzione in cinese, cioè in binario, con tutte le relative

istruzioni.

Per esempio se noi scrivessimo sulla tastiera la parola **STAMPA**, la CPU sulla RAM in corrispondenza di « **STAMPA** » troverà tutte queste istruzioni:

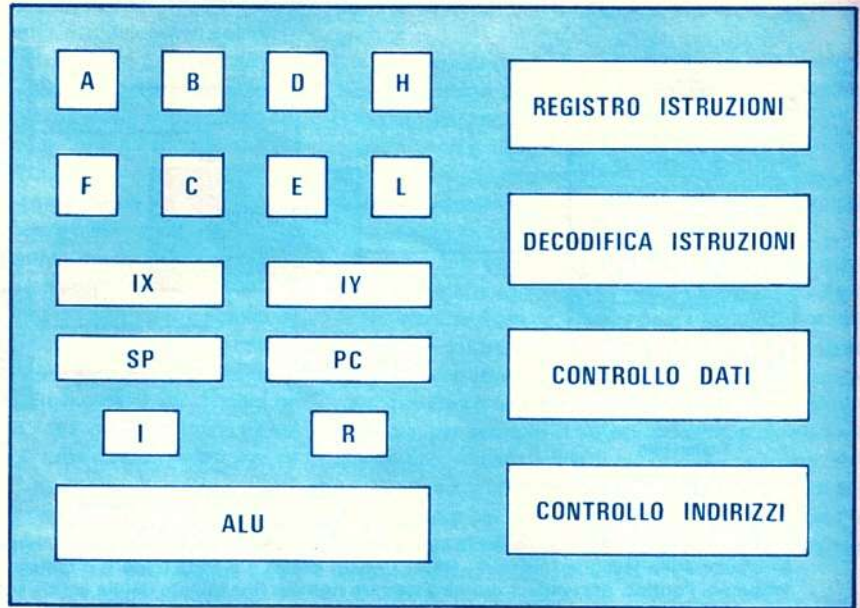
- 1) Prendi i dati scritti in codice binario
- 2) Mandali al traduttore e digli di tradurre questo codice binario in lettere dell'alfabeto dalla A alla Z oppure in numeri decimali
- 3) Di al servitore di portare queste lettere alla stampante
- 4) Ordina alla stampante di eccitare i martelletti corrispondenti a ciascuna lettera
- 5) Quando arriva a fine riga dille di tornare a capo ecc. ecc.;

Se invece sulla tastiera noi scrivessimo **ADDIZIONE**, la CPU nella RAM troverebbe scritte le seguenti istruzioni:

- 1) Chiedi il primo numero e scrivilo sulla lavagna
- 2) Chiedi il secondo numero e scrivilo ancora sulla lavagna
- 3) Fai la somma dei due numeri e scrivila sulla lavagna
- 4) Se dopo questa istruzione trovi **STAMPA**, manda il risultato alla stampante e fallo stampare
- 5) Se invece trovi scritto **VIDEO** fa vedere il risultato sullo schermo del televisore.

Come vedete, anche aggiungendo delle parti nuove al microcomputer, il meccanismo centrale del suo funzionamento non cambia, cioè abbiamo sempre un direttore che prende istruzioni dal suo taccuino oppure dall'esterno e le esegue una per una con precisione assoluta, senza

Fig. 3 Per consentire alla CPU Z80 di svolgere tutte le funzioni richieste, all'interno di questo integrato sono presenti un qualcosa come 10.000 transistor che realizzano i « blocchi » indicati qui a lato.



prendere alcuna iniziativa che non gli sia stata espressamente indicata.

RITORNIAMO alla CPU

Come avrete avuto modo di comprendere dalla lunga chiacchierata fatta in precedenza, il « direttore » di tutto il nostro microcomputer è la C.P.U., cioè l'integrato Z80, il quale riceve i dati e le istruzioni dalla memoria e dalla tastiera esadecimale, li elabora e li manda quindi nuovamente alla memoria oppure al display dove possono essere letti dall'operatore.

Ovviamente sarebbe inutile fornirvi lo schema elettrico interno di questo integrato, in quanto risulta di una complessità veramente eccezionale (pensate che contiene circa 10.000 transistor), tuttavia non possiamo neanche sorvolare totalmente sulla sua costituzione e sui « blocchi » fondamentali che lo costituiscono, diversamente non potreste capire il perché di determinate istruzioni che avremo modo di esaminare più dettagliatamente in seguito.

In pratica all'interno della CPU sono contenuti 6 blocchi fondamentali che potremo così riassumere:

- 1) Un registro per le istruzioni
- 2) 22 registri per i dati di cui 16 ad uso generale e 6 specializzati
- 3) Una decodifica per le istruzioni
- 4) Una unità logico-aritmetica (ALU)
- 5) Una unità di controllo per il BUS dei dati
- 6) Una unità di controllo per il BUS degli indirizzi

Vediamo ora di esaminare più dettagliatamente che co-

sa sono in pratica e quali funzioni svolgono nella vita del microcomputer questi blocchi che abbiamo appena elencato.

REGISTRI di USO GENERALE

Innanzitutto precisiamo che cosa si intende con la parola « registro »: con questa parola si intende in pratica una piccolissima RAM contenuta nell'interno della CPU sulla quale si può « depositare » un dato che deve essere accantonato momentaneamente nel corso di una elaborazione per poi riprenderlo al momento opportuno senza dover ogni volta ricorrere alla memoria esterna.

In altre parole noi possiamo paragonare questi registri a delle piccolissime lavagne che il « direttore » ha a disposizione all'interno del ristorante per scriverci sopra **un sola riga**, cioè un solo numero binario composto da un massimo di 8 cifre se il registro è a **8 bit**, oppure da un massimo di 16 cifre se il registro è a **16 bit**.

Questi registri, come già accennato in precedenza, si dividono in due categorie: quelli « ad uso generale » e quelli « specializzati ».

In pratica la differenza fra queste due categorie consiste nel fatto che mentre il primo tipo di registri può essere sfruttato dall'utilizzatore per scriverci sopra qualsiasi tipo di dati, per esempio il risultato di un'addizione, di una sottrazione oppure l'indirizzo della cella di memoria RAM su cui si vuole scrivere un certo numero, il secondo tipo è riservato a funzioni particolari come per esempio tenere il conteggio dell'ultima istruzione eseguita oppure ricordarsi determinati indirizzi di memoria che serviranno durante

l'esecuzione del programma.

I registri di uso generale disponibili all'interno della CPU sono in totale 16 e sono suddivisi in due gruppi di 8 registri ciascuno contraddistinti dalle seguenti lettere dell'alfabeto:

A	B	C	D	E	F	H	L
A'	B'	C'	D'	E'	F'	H'	L'

Ciascuno dei due gruppi contiene un « accumulatore » (vedi A e A') cioè un particolare registro in cui viene scritto il risultato di qualsiasi operazione aritmetica o logica eseguita dall'ALU, inoltre ciascun gruppo contiene un registro di FLAG (vedi F e F') sul quale vengono automaticamente memorizzate tutte le informazioni inerenti le operazioni che vengono eseguite, cioè:

- Il segno del numero contenuto nell'accumulatore
- Il risultato di una comparazione
- Il risultato di un test di parità
- Il risultato di un test su un singolo bit di un registro
- L'indicazione dell'operazione che si sta eseguendo, cioè addizione o sottrazione
- L'indicazione di ZERO nell'accumulatore
- L'indicazione di una eventuale condizione di CARRY

Per chi ancora non sapesse cosa significa la parola CARRY potremo dire, tanto per fare un paragone, che in pratica è l'equivalente di « over-range », cioè se il risultato di un'operazione eseguita dall'ALU è un numero composto da 9 bit, dovendo « depositare » questo numero nell'accumulatore (che è un registro a 8 bit), ci sarà un bit che rimane escluso, cioè eccede la portata, quindi per non perdere questo bit, la CPU ne conserverà memoria nel registro dei FLAG, indicando appunto su questo registro una condizione di CARRY.

In altre parole è più o meno la stessa cosa che avviene quando con un frequenzimetro a 4 display, in grado cioè di leggere fino a 9.999 Hz, noi andiamo a misurare una frequenza di 12.350 Hz. In tal caso sui display noi vedremo comparire il numero 2.350 però sulla mascherina frontale si accenderà il led dell'over-range per indicarci appunto che abbiamo un 1 che è andato fuori scala.

Le altre 6 « lavagne » (cioè gli altri 6 registri) di uso generale contenute all'interno della CPU sono invece completamente a disposizione dell'utilizzatore il quale può impiegarle singolarmente, cioè B - C - D - E - H - L, oppure in coppia, cioè BC - DE - HL, in modo da ottenere una riga più lunga composta da $8 + 8 = 16$ bit.

Resta ancora da considerare la funzione svolta dai registri che abbiamo indicato con A' - B' - C' ecc. e che possono essere considerati in pratica come la facciata posteriore delle lavagne A - B - C ecc., cioè A' è il corrispondente di A, B' il corrispondente di B, C' il corrispondente di C e così di seguito.

Perché sono stati inseriti questi registri all'interno della CPU è presto detto.

Supponiamo per esempio che il direttore stia eseguendo tutte le operazioni necessarie per poter servire un cliente ed abbia scritto i dati che gli servono sulle lavagne A-B-C-D-E-F-H-L.

A questo punto entra nel ristorante un cliente che ha priorità assoluta, cioè un cliente che ha prenotato in anticipo e deve essere servito prima degli altri.

È ovvio che il direttore deve servire subito questo però non può dimenticarsi di ciò che aveva fatto in precedenza e per raggiungere questo scopo consulta il suo taccuino, cioè la EPROM, e vi trova scritto: **girare tutte le lavagne.**

Esso provvede pertanto a girare tutte le lavagne dopodiché inizia a servire il nuovo cliente scrivendo i dati che gli servono sulla facciata opposta di queste lavagne.

Quando ha finito di servire questo cliente, il direttore legge di nuovo il suo taccuino e vi trova ancora scritto: **girare tutte le lavagne.**

Esso gira quindi tutte le lavagne e si ritrova automaticamente i dati necessari per terminare di servire il primo cliente.

REGISTRI SPECIALIZZATI

Questi registri, come vedesi in fig.3, sono contraddistinti con le lettere **I - R - IX - IY - SP - PC** e sono ancora delle piccole RAM che vengono impiegate dalla CPU per funzioni particolari, ad esempio:

I = INTERRUPT (registro interruzione)

Abbiamo appena visto che quando nel ristorante cinese entra un cliente che ha priorità assoluta, il lavoro del direttore automaticamente si interrompe per passare ad occuparsi di questo nuovo cliente. Il direttore però non sa che cosa deve fare per servirlo, quindi guarda sulla **lavagna I** e qui vi trova scritto il numero di pagina esatto del suo taccuino personale su cui deve andare a leggere le istruzioni che gli necessitano.

In altre parole se sulla **lavagna I** il direttore trova scritto il numero 5, esso sa immediatamente che le istruzioni che gli necessitano per servire il cliente che ha priorità assoluta sono riportate alla pagina 5 del suo taccuino personale.

R = MEMORY REFRESH (registro di rinfresco)

Le memorie RAM impiegate nel microcomputer sono memorie dinamiche che con il tempo tenderebbero a cancellarsi, quindi è necessario rinfrescare continuamente l'informazione in esse contenuta.

A questo provvede automaticamente la CPU la quale, tra un'operazione e l'altra, si preoccupa appunto di rinfrescare le varie celle di memoria aiutandosi con il registro R.

Tanto per continuare il solito paragone del ristorante cinese potremmo dire che la lavagna su cui si scrivono gli

pigiando il 2 comparirà:

Infine, pigiando ancora lo 0, ci apparirà:

Se noi a questo punto continuassimo a pigiare dei numeri sulla tastiera, per esempio 3 - 5 - 8, sui display vedremmo comparire successivamente:

cioè ogni volta tutto il codice di indirizzo si sposta di una posizione verso sinistra, la prima cifra a sinistra scompare ed al suo posto, sul display di destra, compare l'ultimo numero che abbiamo impostato.

Questa caratteristica ci sarà molto utile nel caso in cui ci sbagliassimo ad impostare il nostro codice infatti per ovviare a questo errore sarà sufficiente riscriverlo in modo corretto tutto di seguito.

Una volta che sui display avremo ottenuto l'esatto indirizzo della riga di memoria, cioè:

per vedere che cosa è scritto in questa riga dovremo semplicemente pigiare

CONTROL - 0

cioè dire in pratica al direttore: « questa è la riga che vogliamo vedere; mostraci il suo contenuto ».

Subito sui due display di destra comparirà un codice esadecimale a due cifre che ovviamente, non avendo ancora inserito nessun programma nella RAM, sarà del tutto casuale, per esempio potrebbe comparirvi:

In questo caso noi potremo dire che nella riga di memoria 0120 è contenuta l'istruzione 0B che come vedremo più avanti significa: « Togli 1 al numero contenuto nei registri BC ».

A questo punto, se ci interessa vedere che cosa è contenuto nella riga successiva, cioè nella riga 0121, dovremo pigiare una seconda volta **CONTROL - 0** (Nota importante: i due tasti debbono essere pigiati contemporaneamente altrimenti non fornirete alla CPU nessuna istruzione) ed automaticamente ci apparirà dai quattro display di indirizzo, i numeri 0121 e sui due display sulla destra il codice esadecimale a due cifre contenuto in questa riga.

Tanto per farvi un esempio, potrà apparirvi:

e questo significa che nella riga 0121 è contenuto il numero esadecimale 17.

Se ora volessimo vedere che cosa è contenuto nella riga 0122 dovremmo ancora pigiare CONTROL - 0 e così di seguito.

Se invece volessimo spostarci in tutt'altra parte della memoria, per esempio alla cella 003C, oppure ritornare indietro di una o due celle, dovremmo procedere come segue:

- 1) Pigiare CONTROL - 1
- 2) Impostare sulla tastiera il nuovo codice di riga, cioè 003C
- 3) Pigiare CONTROL - 0

Così facendo automaticamente vedremo comparire sui display il contenuto della riga 3C che nel nostro caso potrebbe risultare:

cioè nella riga 003C è contenuto il codice esadecimale 58.

Ripetiamo ancora però che questi numeri per adesso sono casuali quindi a voi, rifacendo la stessa prova, potrebbero capitarvi dei codici diversi dai nostri.

SCRIVERE NELLA MEMORIA

Nel paragrafo precedente abbiamo appreso come si possa accedere a una determinata riga della memoria RAM per leggere il codice esadecimale in essa contenuto, cioè vi abbiamo insegnato come si possa leggere **nella lavagna interna** del nostro ristorante cinese per vedere le istruzioni e i dati che ci sono scritti.

Adesso vogliamo fare un altro passo, cioè vogliamo insegnarvi a scrivere voi stessi qualcosa su questa lavagna, per esempio una frase utilizzando le lettere alfabetiche disponibili sulla tastiera, cioè FACCIO 18 CAFFÈ E IO DICO 24, due parole qualsiasi cioè BACO e FEDE oppure una targa automobilistica di Firenze, per esempio FI - 738246.

Supponiamo di scrivere queste cose a partire dalla riga 0100 però è ovvio che chiunque di voi potrebbe scegliersi una diversa riga d'inizio purché tutto ciò che si vuole scrivere risulti compreso fra le righe 0100 e 03F0, diversamente andremo fuori dalla lavagna e ciò che scriveremo non rimarrà memorizzato.

Certamente vi chiederete perché abbiamo scritto 0100 e 03F0 anziché 0000 e 03FF come sarebbe logico aspettarsi trattandosi di una memoria da 1.024 righe.

Per questo riteniamo opportuno informarvi che le prime righe della memoria RAM così come le ultime vengono correntemente impiegate dal CPU per memorizzare dei

Innanzitutto dobbiamo precisarvi quali sono i comandi che avete a disposizione, quindi prendete la vostra tastiera, vedi fig. 4 ed osservatela attentamente.

Noterete che oltre ai 16 tasti che servono per impostare i vari codici esadecimali, ne esistono altri due molto importanti e precisamente il tasto di RESET e il tasto di CONTROL.

Pigiando il tasto di **RESET** noi azzeriamo in pratica il Program Counter quindi obblighiamo il « direttore », cioè la CPU, ad andarsi a leggere la prima riga del suo taccuino personale, cioè della EPROM, in modo tale da predisporre per ricevere gli ordini dei clienti.

Questo tasto dovremo quindi ricordarci di pigiarlo prima di iniziare a scrivere qualsiasi programma, in modo da consentire alla CPU di predisporre internamente per accettare tali istruzioni e per eseguirle in modo corretto.

Precisiamo che ogniqualvolta si pigia il RESET, sui display compare:



Il tasto di **CONTROL** deve invece essere sempre **pigiato insieme** ad uno dei tasti « esadecimali » e per non sbagliarvi nel compiere questa operazione noi vi consiglieremo di pigiare prima il tasto CONTROL con l'indice della mano sinistra, poi, tenendo questo sempre pigiato, pigiare 0-1-2 ecc. a seconda delle vostre esigenze con l'indice della mano destra.

In tal modo, a seconda del numero pigiato sulla tastiera esadecimale, potrete ottenere le seguenti funzioni:

CONTROL - 0 = È in pratica il comando di « ENTER » cioè pigiando questi due tasti noi possiamo trasferire all'interno di una cella di memoria oppure di un registro il numero che risulta visualizzato sui due display di destra. Se in precedenza abbiamo impostato sulla tastiera l'indirizzo di una riga di memoria, pigiando questi due tasti ci verrà visualizzato sui display il contenuto di tale riga.

CONTROL - 1 = Pigiando questi due tasti contemporaneamente noi diciamo in pratica alla CPU che vogliamo accedere alla memoria RAM per leggere o modificare dei dati in essa contenuti; la CPU esaudirà subito il nostro ordine però dovremo fornirgli anche il numero di cella che vogliamo ci venga mostrato e successivamente pigiare CONTROL - 0.

CONTROL - 2 = Pigiando questi due tasti noi diciamo invece alla CPU che vogliamo vedere cosa c'è scritto sui registri interni ed essa ci mostrerà subito il numero contenuto nell'accumulatore A. Pigiando CONTROL - 0 potremo poi vedere cos'è contenuto in F, in B, in C, in D, in E, in L, in H, in SP, in PC poi nuovamente in A, in F ecc.

CONTROL - 3 = Pigiando questi due tasti noi diciamo alla CPU di eseguire passo-passo il programma che abbiamo scritto in RAM cioè pigiandoli una prima volta la CPU eseguirà la prima istruzione di tale programma, pigiandoli una seconda volta eseguirà la seconda istruzione poi la terza e così di seguito.

CONTROL - 4 = Con questo comando noi diciamo alla CPU di eseguire tutto il programma che abbiamo scritto in RAM dalla prima istruzione fino allo STOP senza alcuna interruzione.

CONTROL - 5 = Con questa istruzione noi diciamo alla CPU di memorizzare su nastro magnetico il programma contenuto sulla RAM in modo da poterlo conservare per future applicazioni.

CONTROL - 6 = Con questo comando noi diciamo invece alla CPU di leggere dalla cassetta il programma in essa contenuto e di trasferirlo sulla memoria RAM.

CONTROL - 7 = Riservato a future espansioni.

Sulla stessa scheda abbiamo pure disponibili 8 display i quali ci saranno molto utili per poter controllare ciò che scriviamo con la tastiera oppure per controllare i contenuti dei vari registri o celle di memoria.

Come già detto, pigiando il tasto RESET, sui display compare:



e questo significa che il microcomputer è pronto per ricevere i nostri ordini.

La stessa cosa accade anche pigiando CONTROL 1 per accedere alla memoria, infatti il microcomputer resta sempre in attesa che gli forniamo il numero della prima riga che vogliamo esplorare.

Precisiamo che le righe di memoria attualmente disponibili sul microcomputer sono in totale 1.024 e poiché queste, come al solito, vengono numerate partendo da 0, i codici esadecimali che utilizzeremo per individuarle risulteranno compresi fra 0000 (cioè 0) e 03FF (cioè 1.023).

Supponiamo ora di voler vedere che cosa è contenuto nella riga 0120 della RAM.

Per far questo, dopo naturalmente aver pigiato CONTROL - 1, dovremo impostare sulla tastiera esadecimale il numero 0120 in modo da far comparire sui display la scritta:



In pratica, pigiando il primo 0, sui display comparirà:



pigiando l'1 comparirà:



ordini dei clienti pian piano si sporca a tal punto da rendere illeggibile ciò che vi è stato scritto.

Il direttore, per evitare che questo succeda, compie pertanto un'operazione molto semplice ma anche molto utile.

Fra un'istruzione e l'altra si avvicina alla lavagna, cancella per bene la prima riga, la riscrive, poi si segna sul registro R un 1 per ricordarsi appunto quale riga ha cancellato.

Esegue una seconda istruzione poi torna alla lavagna, guarda nel registro R e trovandovi scritto un 1 si ricorda che prima ha cancellato la riga n. 1, quindi ora deve cancellare e riscrivere la n. 2.

Al termine di questa operazione scriverà ovviamente un 2 nel registro R e contemporaneamente passerà ad eseguire un'altra istruzione.

IX - IY = REGISTRI INDICE

Questi due registri entrambi a 16 bit, completamente indipendenti fra di loro, ci permetteranno di realizzare un particolare tipo di indirizzamento alla memoria che viene detto « indicizzato ». In pratica l'uso dei registri indice si rivelerà molto utile quando vorremo memorizzare nelle RAM delle tabelle di dati, comunque su questo argomento torneremo più dettagliatamente in successivi articoli.

SP = STACK-POINTER (Puntatore della catasta)

Quando la CPU sta eseguendo un programma ed avviene un'interruzione, per esempio entra nel « ristorante » un cliente che ha priorità assoluta, abbiamo visto che essa passa immediatamente a servire questo cliente « importante » però quando ha finito deve ricordarsi a che punto era arrivata col cliente precedente per poter ricominciare a servirlo.

Per ottenere questo il direttore si segna in un punto qualsiasi della lavagna, cioè della RAM, tutte le annotazioni che gli servono però quando deve andare a riprendersi queste annotazioni non sa più dove sono e per ritrovarle ha bisogno del registro SP che gli ricordi il punto esatto della lavagna o del taccuino su cui troverà la prossima istruzione da eseguire.

PC = PROGRAM COUNTER

È un registro a 16 bit sul quale noi, prima di far eseguire alla CPU qualsiasi programma, dovremo scrivere il numero della riga su cui essa troverà la prima istruzione che deve essere eseguita. La CPU provvederà poi automaticamente ad aumentare il numero contenuto nel PC man mano che eseguirà le varie istruzioni, in modo da potersi ricordare la riga a cui è arrivata.

Se ad esempio sulla terza riga della lavagna la CPU trovasse scritta un'istruzione di questo genere:

« Salta all'istruzione contenuta nella riga 67 ». automaticamente il numero contenuto nel Program-Counter verrà sostituito con 67, quindi l'esecuzione del programma, anziché proseguire con la riga 4, proseguirà con la riga 67-68-69 ecc.

UNITÀ LOGICO ARITMETICA - ALU

Tutte le operazioni aritmetiche e logiche che ci può fornire la CPU vengono eseguite materialmente dall'ALU.

Essa comunica con l'accumulatore e con gli altri registri interni tramite il BUS-INTERNO-DATI.

Le operazioni che l'ALU può eseguire sono le seguenti:

- Somma
- Sottrazione
- AND-OR-OR ESCLUSIVO
- Confronto
- Rotazione o scorrimento
- Incremento unitario
- Decremento unitario
- Controllo di un singolo bit
- Attivazione di un singolo bit
- Azzeramento di un singolo bit

REGISTRO ISTRUZIONI E CONTROLLO CPU

È in pratica il registro che provvede a interpretare le varie istruzioni che noi forniamo alla CPU in codice esadecimale ed a farle quindi eseguire con assoluta precisione.



Fig. 4 Sulla sinistra della tastiera esadecimale sono presenti due tasti aggiuntivi: il primo, cioè quello di RESET, ci servirà per sbloccare il computer ad esempio dopo che è stato eseguito un HALT; il secondo, cioè quello di CONTROL, pigiato insieme ad un tasto esadecimale, ci permetterà invece di selezionare di volta in volta le varie funzioni disponibili.

ED ORA IMPARIAMO AD USARLO

Precisiamo subito che oggi non vi proporremo cose difficili in quanto lo scopo di questo articolo è insegnare qualcosa a chi non ha mai usato un microcomputer, non a chi lo impiega quotidianamente già da vario tempo.

Proprio per questo vi faremo solo degli esempi molto semplici cioè vi insegneremo, come ad un bambino neonato, a fare i primi passi, per non sbattere la testa contro lo spigolo della tavola, e solo quando sarete più esperti, cioè avrete preso confidenza con la macchina, vi insegneremo anche a fare i salti e le capriole.

dati forniti dalla EPROM, durante l'esecuzione del programma MONITOR, quindi non è possibile utilizzarle per i nostri programmi e le nostre istruzioni.

Per evitare di entrare in questi « campi » riservati alla CPU, vi consigliamo pertanto di « limitare » le righe della RAM su cui scriverete le vostre istruzioni agli indirizzi compresi fra 0100 e 03F0.

Nel nostro caso abbiamo scelto come prima riga la 0100 quindi le operazioni che dovrete compiere saranno nell'ordine:

- 1) Accendete il microcomputer
- 2) Pigiare i tasti CONTROL - 1 (contemporaneamente) in modo da far comparire sui display:

- 3) Scrivete l'indirizzo della prima riga (cioè 0100) sulla tastiera esadecimale e controllate che sui display compaia la scritta:

Se per caso vi foste sbagliati, tornate a scrivere il numero finché non compare questa scritta

- 4) Pigiare ora CONTROL - 0 proprio come facevate quando volevate vedere il contenuto di una riga di memoria.

- 5) Sui display potrebbe comparirvi ad esempio:

cioè nella riga 100 potrebbe essere contenuto ad esempio il codice esadecimale 56 ma poiché questo è un numero casuale potrebbe comparirvi anche FF oppure 00 - 06 - 3C ecc.

Non preoccupatevi comunque di questo in quanto vi abbiamo già anticipato che ogni qualvolta si accende il microcomputer nella memoria RAM entrano delle informazioni casuali che noi ora dovremo cancellare per inserirvi quelle che ci interessano.

La prima informazione che vogliamo inserire è la frase:

FACCIO 18 CAFFE E IO DICO 24

e poiché al massimo in ogni riga possiamo scrivere due « simboli », dovremo prima suddividere questa frase nel modo seguente:

FA-CC-IO-18-CA-FF-EE-IO-DI-CO-24

poi inizieremo ad inserire questi codici.

- 6) Impostate sulla tastiera esadecimale il codice FA in modo da veder comparire sui display la scritta:

- 7) Pigiare CONTROL - 0 in modo da trasferire quanto scritto sui display all'interno della memoria; così facendo passerete anche automaticamente alla cella successiva, cioè sui display comparirà il numero di riga 0101 con di

fianco un codice casuale, per esempio:

- 8) Scrivete sulla tastiera esadecimale CC ed automaticamente sui display vedrete apparire CC al posto di 83, vale a dire:

- 9) Pigiare CONTROL - 0 per trasferire questo dato sulla memoria ed automaticamente passerete alla riga successiva, cioè alla 0102 in cui troverete ancora un numero casuale, per esempio:

- 10) In questa cella dovete memorizzare un IO che otterrete facilmente pigiando i tasti 1 e 0, cosicché sui display vi apparirà:

- 11) Pigiare CONTROL - 0 per trasferire questa informazione alla memoria ed automaticamente passerete alla riga successiva in cui troverete il solito numero casuale, per esempio:

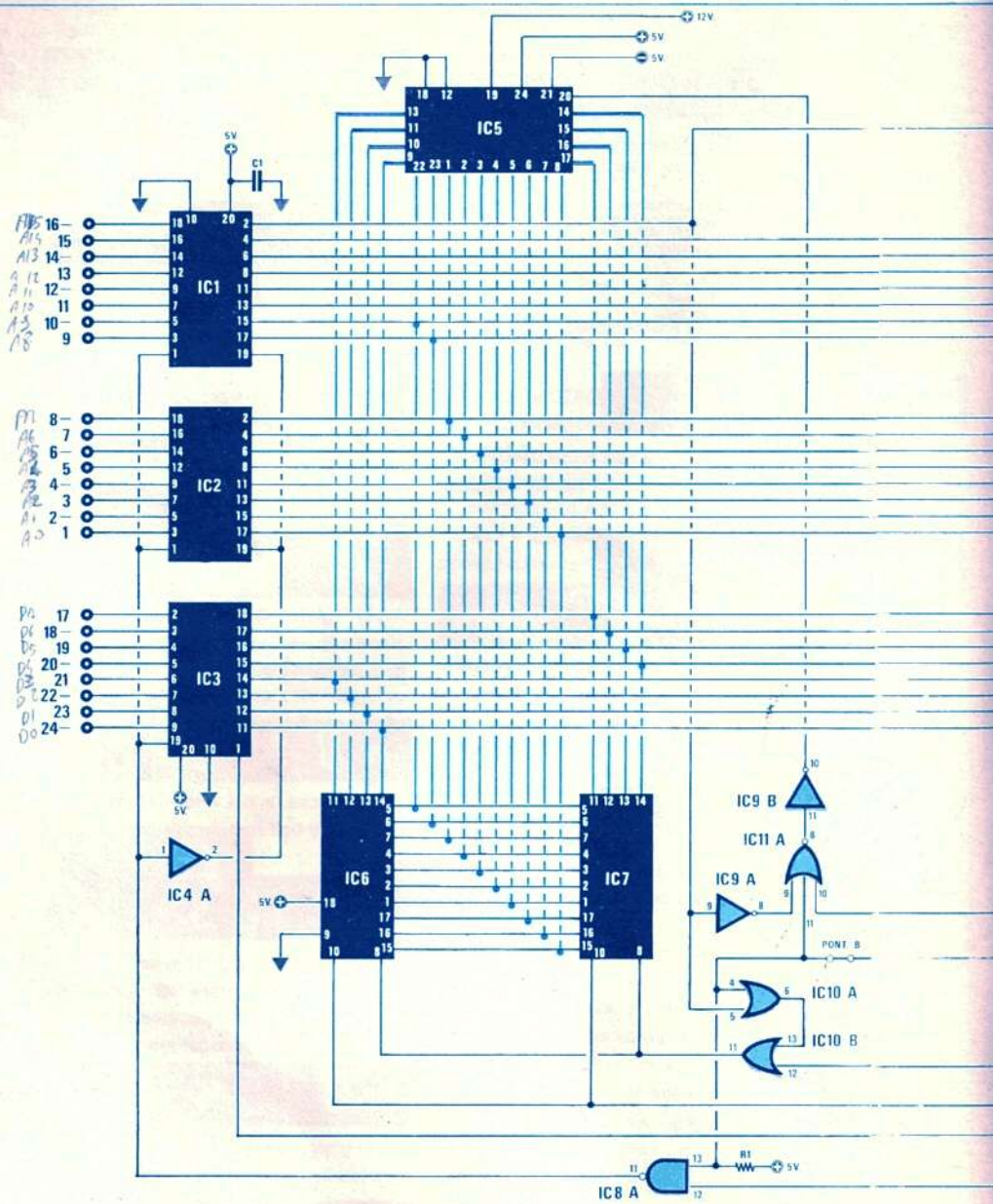
- 12) Poiché a noi interessa scrivere su questa riga il numero 18 batteremo sulla tastiera 18 e sui display comparirà:

- 14) Per trasferire quanto scritto sui display in memoria dovremo ancora pigiare CONTROL - 0 e in questo modo avanderemo di una riga, cioè passeremo alla riga 104 in cui troveremo ancora dei numeri casuali, per esempio:

- 15) Imposteremo ora sulla tastiera esadecimale le altre due lettere, cioè CA in modo da far apparire sui display:

- 16) Per trasferire questo dato in memoria dovremo ora pigiare CONTROL - 0 e così facendo passeremo anche automaticamente alla riga successiva, cioè vedremo comparire sui display per esempio:

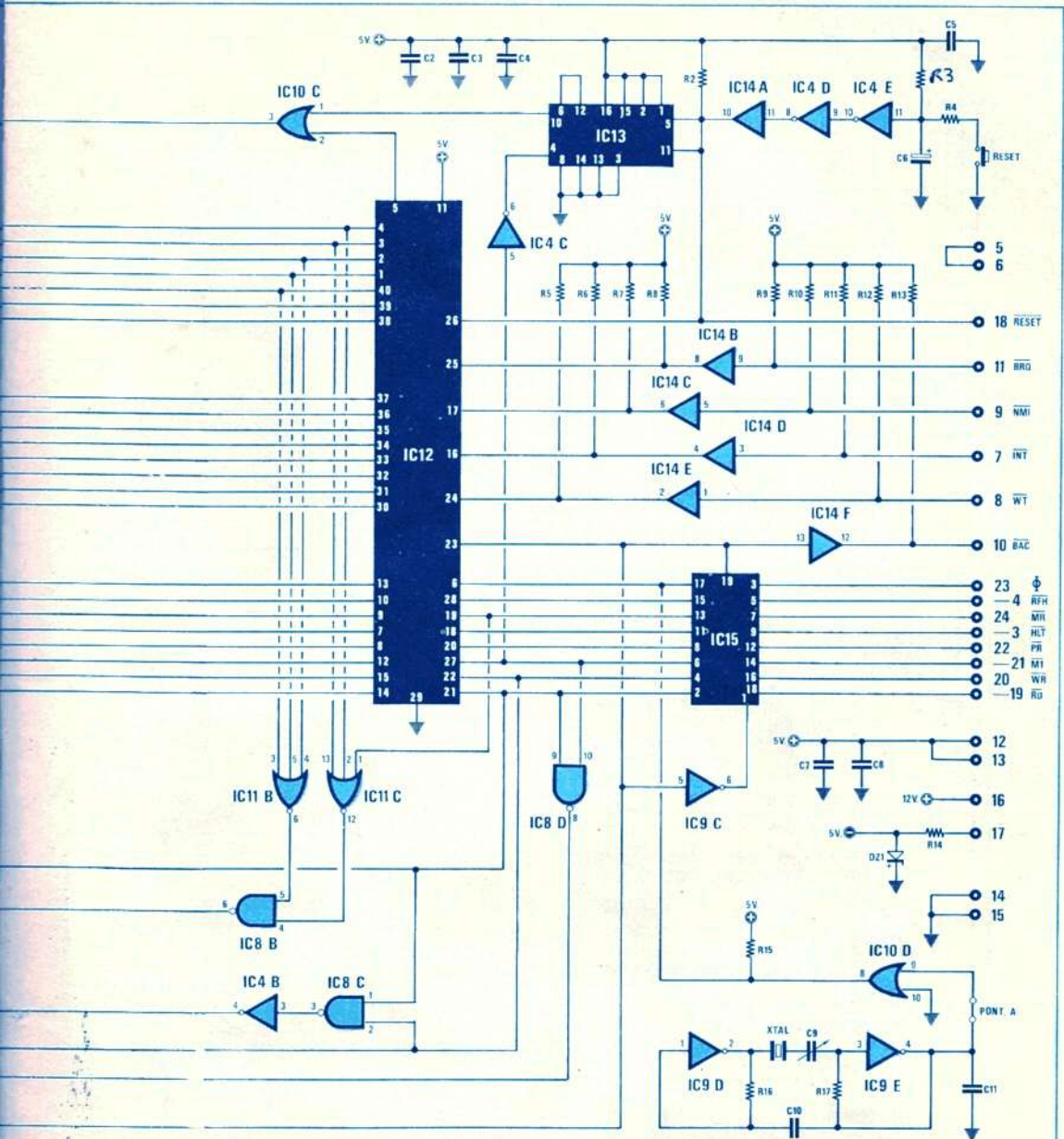
A questo punto ci sembra inutile continuare la spiegazione perché crediamo che ormai tutti avrete capito la procedura che dovete seguire per scrivere nella memoria



COMPONENTI

- R1 = 3.300 ohm 1/4 watt
- R2 = 3.300 ohm 1/4 watt
- R3 = 10.000 ohm 1/4 watt
- R4 = 100 ohm 1/4 watt
- R5 = 3.300 ohm 1/4 watt
- R6 = 3.300 ohm 1/4 watt
- R7 = 3.300 ohm 1/4 watt
- R8 = 3.300 ohm 1/4 watt
- R9 = 3.300 ohm 1/4 watt
- R10 = 3.300 ohm 1/4 watt
- R11 = 3.300 ohm 1/4 watt
- R12 = 3.300 ohm 1/4 watt

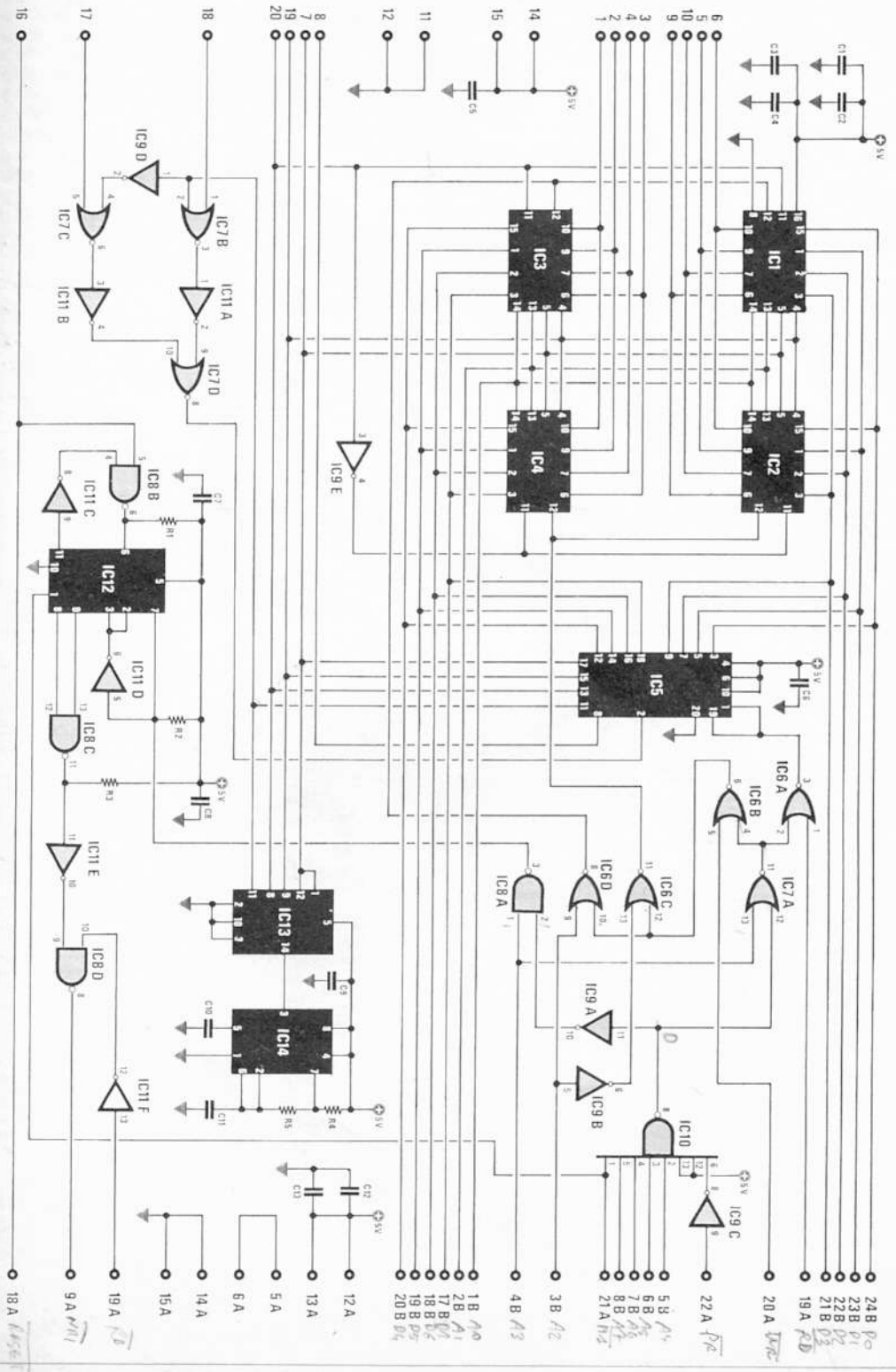
Fig. 5 Schema elettrico relativo alla scheda della CPU. I numeri riportati sui terminali di sinistra e di destra indicano il piedino del connettore a cui ciascun filo risulta collegato. Precisiamo che i terminali di destra si riferiscono al connettore A mentre quelli di sinistra al connettore B.



R13 = 3.300 ohm 1/4 watt
 R14 = 100 ohm 1/4 watt
 R15 = 330 ohm 1/4 watt
 R16 = 680 ohm 1/4 watt
 R17 = 680 ohm 1/4 watt
 C1 = 47.000 pF a disco
 C2 = 47.000 pF a disco
 C3 = 47.000 pF a disco
 C4 = 47.000 pF a disco
 C5 = 47.000 pF a disco
 C6 = 47 mF elettr. 25 volt
 C7 = 47.000 pF a disco

C8 = 47.000 pF a disco
 C9 = 10/40 pF compensatore
 C10 = 1.000 pF a disco
 C11 = 560 pF a disco
 DZ1 = diodo zener 5,1 volt 1 watt
 XTAL = quarzo da 1,920
 IC1 = integrato tipo SN74LS241
 IC2 = integrato tipo SN74LS241
 IC3 = integrato tipo SN74LS45
 IC4 = integrato tipo SN7414
 IC5 = EPROM tipo 2708
 IC6 = integrato tipo 2114

IC7 = integrato tipo 2114
 IC8 = integrato tipo SN74LS00
 IC9 = integrato tipo SN7432
 IC10 = integrato tipo SN7432
 IC11 = integrato tipo SN74ALS27
 IC12 = CPU tipo Z80
 IC13 = integrato tipo SN74LS109
 IC14 = integrato tipo SN7407
 IC15 = integrato tipo SN74LS241



18 A 1856

COMPONENTI

R1 = 3.300 ohm 1/4 watt
 R2 = 3.300 ohm 1/4 watt
 R3 = 3.300 ohm 1/4 watt
 R4 = 100.000 ohm 1/4 watt
 R5 = 22.000 ohm 1/4 watt
 C1 = 47.000 pF a disco
 C2 = 47.000 pF a disco
 C3 = 47.000 pF a disco
 C4 = 47.000 pF a disco
 C5 = 47.000 pF a disco
 C6 = 47.000 pF a disco

C7 = 47.000 pF a disco
 C8 = 47.000 pF a disco
 C9 = 47.000 pF a disco
 C10 = 10.000 pF poliestere
 C11 = 10.000 pF poliestere
 C12 = 47.000 pF a disco
 C13 = 47.000 pF a disco
 IC1 = integrato tipo SN74LS670
 IC2 = integrato tipo SN74LS670
 IC3 = integrato tipo SN74LS670
 IC4 = integrato tipo SN74LS670

IC5 = integrato tipo SN74LS240
 IC6 = integrato tipo SN74LS32
 IC7 = integrato tipo SN74LS32
 IC8 = integrato tipo SN74LS03
 IC9 = integrato tipo SN74LS04
 IC10 = integrato tipo SN74LS30
 IC11 = integrato tipo SN74LS04
 IC12 = integrato tipo SN74LS90
 IC13 = integrato tipo SN74LS93
 IC14 = integrato tipo NE.555

Fig. 6 Schema elettrico relativo all'interfaccia per la tastiera esadecimale; i numeri riportati accanto ai terminali di sinistra si riferiscono ai piedini del connettore a piattina necessario per il collegamento con la tastiera.

RAM tutte le vostre informazioni.

Vi precisiamo solo che quando arriverete alla riga 0109 e pigierete CONTROL - 0 per memorizzare in questa riga l'informazione che vi interessa, sui display non vi comparirà, come sareste portati a supporre l'indirizzo 0110, cioè:

n 0 1 1 0 - 3 C

bensi vi apparirà il numero 010A seguito da un codice casuale, per esempio

n 0 1 0 A - d 2

Infatti bisogna tener presente che anche i numeri di riga sono codificati in « esadecimale », quindi dopo 0109 si ha 010A, poi 010B, 010C, 010D, 010E, 010F e finalmente 0110 poi 0111 ecc.

Ogni volta che vi appare un nuovo numero di riga, dovrete ovviamente impostare sulla tastiera esadecimale il codice che vi interessa quindi battere CONTROL - 0 per trasferirlo all'interno della memoria. Quando avrete scritto l'ultima informazione, pigiate ancora CONTROL - 0 per memorizzarla, poi pigiate CONTROL - 1 in modo da far comparire sui display la scritta:

n - - - - -

A questo punto, se volete riguardarvi ciò che avete scritto all'interno della memoria, dalla prima riga fino all'ultima, impostate sulla tastiera esadecimale il numero di riga 0100, quindi pigiate CONTROL - 0 per far comparire il contenuto di questa riga sui display.

Ovviamente dovrà apparirvi:

n 0 1 0 0 - F A

perché sono queste le lettere che avete inserito in tale riga ed ammesso che ciò si verifichi, pigiando tante volte di seguito CONTROL - 0 quante sono le righe che avete scritto, sui display dovrete veder apparire una dopo l'altra le seguenti informazioni:

n 0 1 0 0 - F A

n 0 1 0 1 - C C

n 0 1 0 2 - 1 0

n 0 1 0 3 - 1 8

n 0 1 0 4 - C A

n 0 1 0 5 - F F

n 0 1 0 6 - E E

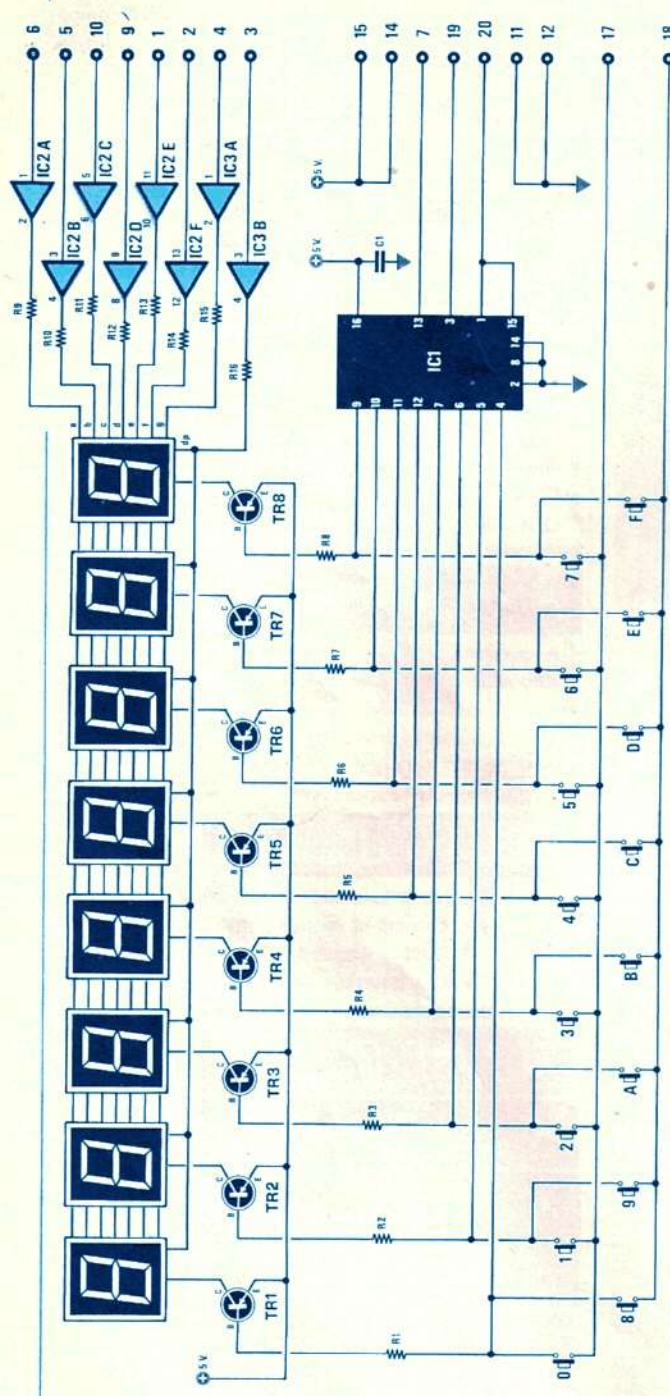


Fig. 7 Schema elettrico della scheda relativa alla tastiera esadecimale e al display di visualizzazione. I tasti 0-8 1-9 2-A ecc. che sembrerebbero collegati fra di loro in parallelo, in realtà non lo sono in quanto prelevano tensione da due fili diversi (dal filo 17 i primi e dal filo 18 i secondi).
I numeri riportati accanto ai terminali di destra si riferiscono sempre al connettore a piattina che collegherà questa scheda alla relativa scheda di interfaccia presente sul BUS.

COMPONENTI

- R1-R8 = 470 ohm 1/4 watt
- R9-R16 = 68 ohm 1/4 watt
- R17 = 100 ohm 1/4 watt
- R18 = 10.00 ohm 1/4 watt
- R19 = 3.300 ohm 1/4 watt
- C1-C3 = 47.000 pF a disco
- C4 = 47 mF elettr. 25 volt

- TR1-TR8 = transistor PNP tipo BC177
- IC1 = integrato tipo SN74LS156
- IC2 = integrato tipo SN7407
- IC3 = integrato tipo SN7407
- IC4 = integrato tipo SN7414
- 8 display anodo comune tipo TIL...321

n0 107-10

n0 108-d1

n0 109-C0

n0 10A-24

n0 10b-00

n0 10C-bA

n0 10d-C0

n0 10E-00

n0 10F-FE

n0 110-dE

n0 111-00

n0 112-F1

n0 113-73

n0 114-82

n0 115-46

Come si potrà notare, per separare la frase FACCIÒ 18 CAFFÈ E IO DICO 24 dalla parola BACO, oppure la parola BACO da FEDE, oppure quest'ultima dalla targa FI738246, abbiamo inserito degli 00, tuttavia ci sembra inutile aggiungere che al posto di questi 00 avremmo potuto mettere indifferentemente anche degli FF in modo da poter distinguere le varie frasi fra loro.

Noterete che nella precedente tabella, accanto ad ogni dato, abbiamo indicato il relativo codice di indirizzo, cioè il numero della riga in cui è contenuto, e questo ci sarà molto utile quando vorremo andare a rileggere tali informazioni.

Supponendo infatti che ci interessi controllare solo l'esattezza del numero di targa è inutile che passiamo in

rassegna tutte le celle dalla 0100 fino alla 0112 per poterlo trovare, bensì sarà sufficiente, dopo aver pigiato CONTROL - 1, impostare sulla tastiera esadecimale il numero 0112 ed a questo punto, pigiando CONTROL - 0, se non abbiamo commesso errori, sui due display di destra dovrà comparire FI, cioè:

n0 112-F1

Pigiando ancora CONTROL - 0 dovrà comparire:

n0 113-73

e così di seguito.

COME CORREGGERE UN ERRORE

Supponiamo che rileggendo quanto avete scritto in precedenza, vi accorgete che per esempio alla riga 010C avete scritto BB invece che BA.

Questo è un errore molto facile da commettere soprattutto quando ci si crede già esperti programmatori e si scrive sulla tastiera senza guardare i display.

Grazie al cielo è però anche un errore molto facile da correggere in quanto per ripristinare la normalità è sufficiente procedere come qui di seguito indicato.

1) Pigiare CONTROL - 1 in modo da far comparire sui display la scritta

n - - - - -

2) Impostate sulla tastiera il numero della riga da correggere (nel nostro caso 010C) e controllate che sui display compaia la scritta:

n0 10C - - -

3) Pigiare CONTROL - 0 ed automaticamente vi apparirà sui display:

n0 10C - bb

4) Impostate sulla tastiera il codice esatto, cioè BA, e quando sui display leggerete:

n0 10C - bA

pigiare ancora CONTROL - 0 per trasferire questo nuovo dato alla memoria.

Se invece vi accorgete dell'errore man mano che esplorate le varie righe, cosa questa molto più probabile, non è necessario pigiare CONTROL - 1, bensì sarà sufficiente impostare il codice corretto sulla tastiera e pigiare quindi CONTROL - 0 per trasferirlo alla memoria.

In questo modo passerete anche automaticamente alla

riga successiva il cui contenuto vi verrà visualizzato sui display subito dopo aver pigiato CONTROL - 0.

PER LEGGERE SUI REGISTRI INTERNI ALLA CPU

Con le istruzioni che fino qui vi abbiamo fornito voi potete già effettuare diverse prove, cioè inserire nella memoria RAM dei dati, andarli a ricercare passo per passo oppure andare direttamente a quello che interessa, modificare dei dati precedentemente inseriti oppure correggerli se sbagliati.

Questo però solo all'interno della memoria RAM perché se per caso voleste andare a leggervi il numero contenuto per esempio nel registro A (cioè nell'accumulatore) oppure nel registro PC (cioè nel Program Counter) non sapreste che pesci pigliare.

In questo paragrafo vi spiegheremo pertanto quali operazioni è necessario compiere per poter leggere, scrivere o modificare un'informazione entro questi registri.

Se noi volessimo leggere il contenuto dei vari registri dovremmo procedere come segue:

1) Pigiare i tasti CONTROL — 2 (ricordatevi sempre di pigiare per primo il tasto CONTROL, poi tenendolo premuto, pigiare di seguito il n. 2).

Automaticamente sui display comparirà una A seguita come avveniva in precedenza da due numeri casuali, per esempio:

Constaterete che le differenze sostanziali rispetto a quanto si vedeva esplorando la memoria RAM sono una « u » al posto di una « n » nel primo display a sinistra e una A al posto del « numero di riga » per indicarci il registro che stiamo esplorando.

In pratica noi potremmo affermare che nel registro A è contenuto il codice esadecimale C0.

2) Per passare al registro successivo dovrete ora pigiare CONTROL — 0 ed automaticamente sui display comparirà la lettera F seguita ancora da due numeri casuali, cioè:

In altre parole, nel registro F dei FLAG è contenuta l'informazione 80.

3) Pigiando ancora CONTROL — 0 passeremo al registro successivo, quindi sui display potrà apparirci:

Questo significa che nel registro B è contenuta l'informazione 64.

4) Dopo il registro B, pigiando ancora CONTROL — 0,

sui display ci verrà visualizzato il contenuto del registro C e qui potranno esserci ancora due numeri casuali, per esempio:

5) Pigiando CONTROL — 0 passeremo ora al registro D quindi sui display vedremo comparire una « d » minuscola (la lettera b e la lettera d, come vedesi nelle figure, sono sempre minuscole) seguita come al solito da due numeri casuali, per esempio:

6) Dopo il registro D, pigiando CONTROL — 0 passeremo al registro E entro cui potrebbe trovarsi ad esempio l'informazione 02, cioè:

7) Pigiando ancora CONTROL — 0 passeremo al registro L ed ammesso che questo contenga per esempio il codice esadecimale 64, sui display vedremo apparire:

8) Resta ancora un registro di uso generale da esaminare e precisamente il registro H il cui contenuto ci verrà visualizzato sui display pigiando ancora una volta CONTROL - 0 e precisamente potrà apparirci:

9) Dopo il registro H, pigiando ancora CONTROL — 0 avremo una sorpresa infatti vedremo comparire la scritta SP (Stack Pointer) seguita da 4 numeri anziché da 2 come avveniva in precedenza.

In genere i numeri che compaiono sono:

tuttavia nel caso non aveste pigiato in precedenza il pulsante di RESET ed aveste fatto eseguire al microcomputer un programma qualsiasi potreste trovare anche in questo registro dei numeri casuali.

10) Pigiando ancora CONTROL — 0 ci verrà infine visualizzato il contenuto del registro PC (cioè del Program Counter) sempre sotto forma di un numero a quattro cifre, per esempio:

11) A questo punto abbiamo esaurito tutta la serie di registri specializzati o di uso generale accessibili dalla tastiera, quindi pigiando ancora CONTROL — 0 ci verrà visualizzato di nuovo il contenuto del registro A, cioè:

Poi vedremo ancora il registro F, B, C, D ecc.

MODIFICARE IL CONTENUTO DI UN REGISTRO

Supponiamo a questo punto che per esigenze di programma ci interessi di modificare i dati contenuti nei registri A-B-PC e precisamente ci interessi inserire in questi registri le seguenti informazioni:

A = FF
B = 3B
PC = 0100

Le operazioni da compiere sono le seguenti:

1) Battere CONTROL — 2 in modo da far comparire nuovamente sui display:

2) Dovendo modificare C0 in FF, scriveremo sulla tastiera esadecimale FF e quando vedremo comparire questa scritta sui due display di destra

pigeremo CONTROL — 0. Automaticamente vedremo comparire:

cioè il contenuto del registro F che è quello immediatamente successivo ad A e poiché a noi non interessa modificare questo contenuto, batteremo ancora CONTROL — 0 per passare al registro B.

4) Sui display vedremo apparire:

e poiché vogliamo sostituire questo codice con 3B, imposteremo sulla tastiera esadecimale 3B

quindi pigeremo CONTROL — 0 per trasferirlo all'interno del registro.

5) A questo punto, poiché ci resta da modificare il solo registro PC, pigeremo tante volte CONTROL — 0 quante sono necessarie per veder comparire sui display il contenuto di tale registro, cioè:

6) Raggiunta tale condizione imposteremo sulla tastiera esadecimale il nuovo codice, cioè 0100,

quindi pigeremo CONTROL — 0 per trasferirlo all'interno del registro.

Se ora volessimo controllare se ciò che abbiamo modificato è entrato veramente nella memoria, non dovremo fare altro che pigiare tante volte CONTROL — 0 come indicato nel paragrafo precedente fino a passare in rassegna tutti i registri.

LE ISTRUZIONI E I DATI

Finora vi abbiamo parlato genericamente di « istruzioni » e di « programma » senza scendere nei dettagli, cioè senza specificarvi esattamente che cosa si intende con queste parole.

Per chiarirvi le idee in proposito, ma crediamo che ormai lo avrete capito anche da soli, vi diremo pertanto che un'istruzione è un ordine singolo che noi diamo al computer, per esempio:

« Carica il numero 33 nel registro B »

oppure

« Trasferisci il contenuto del registro B nel registro C ».

Un programma invece è tutta una serie di ordini (o di istruzioni) che noi diciamo al computer di eseguire nella sequenza indicata.

Precisiamo che pure essendo un'istruzione un ordine singolo, non è detto che questo si debba scrivere nella memoria RAM sempre su una sola riga, infatti come avremo modo di vedere più avanti esistono delle istruzioni che occupano una sola riga di memoria, altre istruzioni che ne occupano due, ed altre ancora che ne occupano tre oppure quattro.

Il motivo di questa differente lunghezza fra un'istruzione e l'altra è presto detto e si può facilmente comprendere ripensando un attimo all'esempio del ristorante cinese che vi abbiamo fatto all'inizio di questo articolo.

Se il cliente che è seduto al tavolo ha finito di mangiare e vuole il conto, basta che dica « CONTO » ed automaticamente verrà esaudito, cioè con una sola parola si è già fatto comprendere.

Se invece il cliente ha ordinato da mangiare e vuole 2 uova sode non gli basterà più una sola parola per farsi comprendere, infatti dovrà prima dire

« UOVA SODE »

poi specificare quante ne vuole, cioè:

« 02 »

diversamente il cameriere non conoscendo la quantità desiderata, potrebbe portargliene una sola oppure 3 o 4.

Se poi questo cliente, oltre a volere le due uova sode, le vuole anche già sbucciate, non saranno più sufficienti due righe per farsi comprendere, bensì ne occorreranno tre e precisamente:

UOVA SODE

SBUCCIATE

02

00	NOP	20	JR NZ,DIS	40	LD B,B	60	LD H,B	80	ADD A,B
01	LD BC,nn	21	LD HL,nn	41	LD B,C	61	LD H,C	81	ADD A,C
02	LD (BC),A	22	LD (nn),HL	42	LD B,D	62	LD H,D	82	ADD A,D
03	INC BC	23	INC HL	43	LD B,E	63	LD H,E	83	ADD A,E
04	INC B	24	INC H	44	LD B,H	64	LD H,H	84	ADD A,H
05	DEC B	25	DEC H	45	LD B,L	65	LD H,L	85	ADD A,L
06	LD B,n	26	LD H,n	46	LD B,(HL)	66	LD H,(HL)	86	ADD A,(HL)
07	RLCA	27	DAA	47	LD B,A	67	LD H,A	87	ADD A,A
08	EX AF,AF'	28	JR Z,DIS	48	LD C,B	68	LD L,B	88	ADC A,B
09	ADD HL,BC	29	ADD HL,HL	49	LD C,C	69	LD L,C	89	ADC A,C
0A	LD A,(BC)	2A	LD (HL),(nn)	4A	LD C,D	6A	LD L,D	8A	ADC A,D
0B	DEC BC	2B	DEC HL	4B	LD C,E	6B	LD L,E	8B	ADC A,E
0C	INC C	2C	INC L	4C	LD C,H	6C	LD L,H	8C	ADC A,H
0D	DEC C	2D	DEC L	4D	LD C,L	6D	LD L,L	8D	ADC A,L
0E	LD C,n	2E	LD L,n	4E	LD C,(HL)	6E	LD L,(HL)	8E	ADC A,(HL)
0F	RRCA	2F	CPL	4F	LD C,A	6F	LD L,A	8F	ADC A,A
10	DJNZ DIS	30	JR NC,DIS	50	LD D,B	70	LD (HL),B	90	SUB B
11	LD DE,nn	31	LD SP,nn	51	LD D,C	71	LD (HL),C	91	SUB C
12	LD (DE),A	32	LD (nn),A	52	LD D,D	72	LD (HL),D	92	SUB D
13	INC DE	33	INC SP	53	LD D,E	73	LD (HL),E	93	SUB E
14	INC D	34	INC (HL)	54	LD D,H	74	LD (HL),H	94	SUB H
15	DEC D	35	DEC (HL)	55	LD D,L	75	LD (HL),L	95	SUB L
16	LD D,n	36	LD (HL),n	56	LD D,(HL)	76	HALT	96	SUB (HL)
17	RLA	37	SCF	57	LD D,A	77	LD (HL),A	97	SUB A
18	JR DIS	38	JR C,DIS	58	LD E,B	78	LD A,B	98	SBC A,B
19	ADD HL,DE	39	ADD HL,SP	59	LD E,C	79	LD A,C	99	SBC A,C
1A	LD A,(DE)	3A	LD A,(nn)	5A	LD E,D	7A	LD A,D	9A	SBC A,D
1B	DEC DE	3B	DEC SP	5B	LD E,E	7B	LD A,E	9B	SBC A,E
1C	INC E	3C	INC A	5C	LD E,H	7C	LD A,H	9C	SBC A,H
1D	DEC E	3D	DEC A	5D	LD E,L	7D	LD A,L	9D	SBC A,L
1E	LD E,n	3E	LD A,n	5E	LD E,(HL)	7E	LD A,(HL)	9E	SBC A,(HL)
1F	RRA	3F	CCF	5F	LD E,A	7F	LD A,A	9F	SBC A,A

In pratica le prime due righe di questa istruzione costituiscono il CODICE OPERATIVO, cioè ci servono per spiegare al direttore o al cameriere l'operazione che deve compiere, mentre la terza riga contiene il DATO che gli necessita per poter compiere in modo corretto tale operazione.

Quindi più un'operazione è complessa, più righe di memoria si richiederanno per poterla scrivere.

Quante sono le istruzioni che possiamo dare al micro-computer?

Anche questa è una domanda più che logica alla quale noi risponderemo che lo Z80 può riconoscere ed eseguire circa 700 istruzioni.

Ovviamente un numero così alto di istruzioni non è facile da ricordare tuttavia non dovete preoccuparvi perché quando vi forniremo il linguaggio BASIC, molti problemi diventeranno di più facile risoluzione.

Precisiamo che ciascuna istruzione è contraddistinta da

un proprio codice esadecimale a due cifre, per esempio C3-F8-50 ecc. che va scritto su una sola riga della memoria, oppure da un codice a 4 cifre, per esempio FD21, che va scritto su due righe successive, cioè:

FD
21

Anche i dati che noi inseriamo dopo il codice operativo possono essere a due oppure a quattro cifre ed in quest'ultimo caso dovranno essere scritti su due righe successive, tenendo presente una regola molto importante e cioè che nella prima riga vanno sempre scritte le due cifre sulla destra di tale numero mentre nella seconda riga dovremo scrivere le due cifre che stanno sulla sinistra.

Per esempio dovendo inserire come « dato » dopo un codice operativo qualsiasi, il numero **0345**, scriveremo

45 sulla prima riga
03 sulla seconda riga

A0 AND B	C0 RET NZ	E0 RET PO
A1 AND C	C1 POP BC	E1 POP HL
A2 AND D	C2 JP NZ,nn	E2 JP PO,nn
A3 AND E	C3 JP nn	E3 EX (SP),HL
A4 AND H	C4 CALL NZ,nn	E4 CALL PO,nn
A5 AND L	C5 PUSH BC	E5 PUSH HL
A6 AND (HL)	C6 ADD A,n	E6 AND n
A7 AND A	C7 RST 0	E7 RST 20H
A8 XOR B	C8 RET Z	E8 RET PE
A9 XOR C	C9 RET	E9 JP (HL)
AA XOR D	CA JP Z,nn	EA JP PE nn
AB XOR E	CC CALL Z,nn	EB EX DE,HL
AC XOR H	CD CALL nn	EC CALL PE,nn
AD XOR L	CE ADC A,n	EE XOR n
AE XOR (HL)	CF RST 8	EF RST 28H
AF XOR A		
B0 OR B	D0 RET NC	F0 RET P
B1 OR C	D1 POP DE	F1 POP AF
B2 OR D	D2 JP NC,nn	F2 JP P,nn
B3 OR E	D3 OUT (n),A	F3 DI
B4 OR H	D4 CALL NC,nn	F4 CALL P,nn
B5 OR L	D5 PUSH DE	F5 PUSH AF
B6 OR (HL)	D6 SUB n	F6 OR n
B7 OR A	D7 RST 10H	F7 RST 30H
B8 CP B	D8 RET C	F8 RET M
B9 CP C	D9 EXX	F9 LD SP,HL
BA CP D	DA JP C,nn	FA JP M,nn
BB CP E	DB IN A,(n)	FB EI
BC CP H	DC CALL C,n	FC CALL M,nn
BD CP L	DE SBC A,n	FE CP n
BE CP (HL)	DF REST 18H	FF RST 38H
BF CP A		

Nella tabella qui sopra riportata il lettore troverà elencati i codici esadecimale relativi alle più importanti istruzioni dello Z80 e di fianco a ciascuno di essi il corrispondente « mnemonico », cioè una frase abbreviata che ci sarà molto utile per ricordarci a mente il significato di ciascuna istruzione.

Facciamo presente che questa tabella comprende solo la prima parte delle istruzioni dello Z80 e precisamente le più semplici nonché quelle di uso più comune, perché quelle più complesse vi verranno presentate in altre tabelle che pubblicheremo sui prossimi numeri.

Ovviamente quelli che noi abbiamo chiamato « mnemonici », cioè LD A,B oppure ADD A,C saranno ancora per voi delle frasi incomprensibili, cioè delle frasi in « cinese », quindi cercheremo di aiutarvi ad interpretarle fornendovi qui di seguito una specie di dizionarietto di tali termini con la relativa spiegazione.

**TABELLA n. 1
dei MNEMONICI**

ADC	=	addiziona con CARRY
ADD	=	addiziona senza CARRY
AND	=	esegui la funzione AND
BIT	=	controlla un bit di un registro
CALL	=	vai a una subroutine
CCF	=	complementa il flag di CARRY
CP	=	confronta un numero con un altro
DAA	=	trasforma in decimale il contenuto del registro A
DEC	=	decrementa, cioè toglie 1 al contenuto di un registro
EX	=	scambia un registro con l'alternativo
HALT	=	fermati
IM	=	seleziona un tipo di interrupt
IN	=	ingresso di dati
INC	=	incrementa, cioè aggiungi 1 al contenuto di un registro
JP	=	salta alla riga di memoria indicata
JR	=	salta un certo numero di righe di memoria
LD	=	carica un numero in un registro o in una cella di memoria
NEG	=	cambia segno al contenuto dell'accumulatore
NOP	=	non operare
OR	=	esegui la funzione OR
OUT	=	uscita dati dalla memoria
POP	=	riprendi dallo STACK
PUSH	=	carica nello STACK
RES	=	azzerà un bit di un registro
RET	=	ritorna da una subroutine
RL	=	ruota verso sinistra
RR	=	ruota verso destra
RST	=	torna al monitor
SBC	=	sottrai con CARRY
SCF	=	poni il bit C del registro F uguale a 1
SET	=	poni un bit di un registro uguale a 1
SL	=	shifta verso sinistra
SR	=	shifta verso destra
SUB	=	sottrai senza CARRY
XOR	=	esegui la funzione OR-ESCLUSIVO

DIZIONARIETTO DEI MNEMONICI

LD = È l'abbreviazione della parola inglese LOAD e tradotto in italiano significa « CARICA », cioè carica un determinato numero in un registro oppure in una riga della memoria RAM.

Ovviamente dopo LD bisogna sempre specificare che cosa si vuol caricare e dove lo si vuol caricare; per esempio:

LD A,35 significa « carica il numero 35 nel registro A »

LD B,C significa « carica il contenuto del registro C sul registro B »

LD (0100),A significa « carica il contenuto dell'accumulatore A nella riga di memoria 0100 ».

Quindi ciò che troviamo scritto subito dopo LD è il registro o la cella di memoria in cui vogliamo che vada a finire il dato, mentre quello che troviamo **dopo la virgola** è il dato oppure il registro che contiene attualmente questo dato.

EX = È l'abbreviazione di EXCHANGE cioè « SCAMBIA » e viene utilizzata quando si vuol dire al direttore del ristorante di girare le sue lavagne, cioè quando si vogliono utilizzare i registri alternativi A'-B'-C' ecc. al posto di A-B-C-D-E-F-H-L o viceversa.

CP = È l'abbreviazione della parola inglese « COMPARE » cioè « CONFRONTA » fra di loro due registri e dicci se il loro contenuto è uguale, oppure uno contiene un numero maggiore dell'altro.

ADD = Significa letteralmente « ADDIZIONE », cioè addiziona al contenuto di un registro un determinato numero oppure il contenuto di un altro registro.

ADC = Significa letteralmente « ADDIZIONE con CARRY », cioè se l'ultima addizione eseguita ha dato luogo ad un CARRY, vale a dire ad un numero maggiore di 8 bit, al risultato di questa nuova addizione il calcolatore aggiunge un 1. In pratica il calcolatore fa la stessa cosa che facciamo noi quando eseguiamo l'addizione in colonna dei numeri 8 e 5, infatti essendo il risultato 13, cioè un numero a due cifre, in questa colonna scriviamo 3 e riportiamo l'1 nella colonna successiva addizionandolo quindi ad eventuali altri numeri ivi presenti.

SUB = È l'abbreviazione della parola « SUBTRACT » cioè « SOTTRAI » al contenuto di un registro un certo numero oppure anche il contenuto di un altro registro.

SBC = È in pratica il corrispondente di ADC e significa « SOTTRAI con CARRY »; questa istruzione ci sarà molto utile quando dovremo eseguire delle sottrazioni di numeri a 16 oppure a 32 bit.

AND = Significa « Esegui la funzione AND » sul contenuto di due registri o su due numeri qualsiasi.

OR = Significa « Esegui la funzione OR » sul contenuto di due registri o su due numeri qualsiasi.

XOR = Significa « Esegui la funzione OR ESCLUSIVO » sul contenuto di due registri o su due numeri qualsiasi.

Nota: sul significato di queste ultime tre istruzioni tor-

neremo più dettagliatamente in successivi articoli.

INC = È l'abbreviazione della parola inglese « INCREMENT » e significa letteralmente « ADDIZIONE 1 » al contenuto di un registro o di una cella di memoria puntata da un registro.

DEC = È l'abbreviazione della parola inglese « DECREMENT » e significa letteralmente « SOTTRAI 1 » al contenuto di un registro oppure di una cella di memoria.

RL = È l'abbreviazione della parola inglese « ROTATE LEFT » e significa letteralmente « RUOTA VERSO SINISTRA » il contenuto di un registro o di una cella di memoria.

RR = È la traduzione della parola inglese « ROTATE RIGHT », cioè « RUOTA verso DESTRA » il contenuto di un registro o di una cella di memoria.

SL = È l'abbreviazione della parola inglese « SHIFT LEFT » cioè « FAI SLITTARE DI UN POSTO VERSO SINISTRA » tutti i bit di un numero contenuto in un registro o in una cella di memoria.

SR = È l'abbreviazione di « SHIFT RIGHT » e al contrario del precedente significa « FAI SLITTARE DI UNA POSIZIONE VERSO DESTRA ».

DAA = È l'abbreviazione di « DECIMAL ADJUST ACCUMULATOR » cioè « TRASFORMA IN DECIMALE IL CONTENUTO DELL'ACCUMULATORE »; questa istruzione ci servirà per convertire da esadecimale a « decimale » il risultato di un'addizione o una sottrazione contenuto nell'accumulatore.

CPL = Con questa istruzione noi diciamo alla CPU di fare il « complemento » del numero contenuto nell'accumulatore, cioè di convertire tutti gli 1 in 0 e tutti gli 0 in 1.

NEG = È l'abbreviazione della parola inglese « NEGATE » e significa in pratica « CAMBIA SEGNO » al numero contenuto nell'accumulatore, cioè se è positivo trasformalo in negativo e viceversa.

NOP = Equivale a « NON OPERARE » cioè salta questa riga e passa a quella successiva.

HALT = Fermati, il programma è finito.

IM = È l'abbreviazione della parola inglese « INTERRUPT MODE » e serve per selezionare il tipo di risposta che la CPU deve dare a un'eventuale richiesta di interruzione del programma da parte di una periferica.

BIT = Serve per vedere se un determinato « bit » di un registro è uguale a 1 oppure uguale a 0.

SET = Pone un bit di un registro in condizione 1.

RES = Pone un bit di un registro in condizione 0.

JP = È l'abbreviazione della parola inglese « JUMP » e significa letteralmente « SALTA », cioè salta direttamente alla riga indicata subito dopo JP tralasciando di eseguire tutte le istruzioni che si trovano nel mezzo.

JR = È sempre un salto come il precedente solo che questa volta noi non specifichiamo la riga a cui deve andare, bensì gli diciamo solo di quante righe deve spostarsi rispetto all'ultima istruzione che ha eseguito.

CALL = Vai a una « subroutine » (si legge sabrutin)

cioè a un gruppo di istruzioni che essendo richiamate molte volte nel corso del programma sono state scritte in un angolo della lavagna mettendole bene in evidenza.

RET = La subroutine o qualsiasi altra interruzione è finita; ritorna ad eseguire il normale programma ripartendo dall'istruzione successiva a quella su cui ti eri fermato.

IN = Ingresso dati dalle periferiche verso la memoria.

OUT = Uscita dati dalla memoria verso le periferiche.

POP = Carica nel registro IX o IY (o in un'altra coppia di registri) il numero contenuto nella riga più bassa dello STACK (vedremo in seguito più dettagliatamente cosa significa questo discorso).

PUSH = Carica il contenuto di IX-IY o di un'altra coppia di registri dentro lo STACK.

SCF = Poni il bit C del registro F uguale a 1.

DI = Disabilita gli « interrupts ».

EI = Abilita gli « interrupts ».

DJNZ = Diminuisci di 1 il contenuto del registro B e quando B = 0 esegui un « salto » di X istruzioni.

RST = È simile all'istruzione CALL, cioè è sempre un salto ad una « subroutine » contenuta però questa volta nella parte bassa della memoria.

ED ORA SCRIVIAMO UN PROGRAMMA

Adesso che abbiamo visto come si può leggere e scrivere all'interno della memoria RAM oppure dei vari registri contenuti nella CPU non possiamo certo affermare di essere già degli esperti conoscitori del microcomputer, però possiamo dire che il primo scoglio lo abbiamo superato.

Tanto per fare un esempio potremmo paragonarci ad un allievo di una scuola guida che ha imparato a inserire le quattro marce più la retromarcia, però non ha ancora provato a partire ed è proprio questo che stiamo per fare, cioè siamo in procinto di far eseguire il nostro primo programma al calcolatore.

Come primo tentativo è ovvio che non cercheremo di fargli fare dei « salti mortali » proprio come l'allievo automobilista non partirà subito ai 100 all'ora, bensì cercheremo di fargli eseguire un paio di semplicissime istruzioni.

In pratica il programma che vogliamo far eseguire al microcomputer è questo:

noi scriviamo con la tastiera un dato nel registro B (per esempio C5) quindi vogliamo che la CPU lo trasferisca automaticamente nel registro A, nel registro C e nel registro D.

Per ottenere questo sono sufficienti 4 istruzioni e precisamente:

78 = LD A,B (carica B su A)

48 = LD C,B (carica B su C)

50 = LD D,B (carica B su D)

FF = Stop (Fermati)

che scriveremo in memoria, a partire dalla riga n. 0100 nel modo seguente:

1) Pigiare CONTROL — 1 in modo da far apparire sui display:

2) Impostate sulla tastiera il numero della riga da cui volete partire, cioè 0100 e quando sui display leggerete:

battete CONTROL — 0.

3) Sui due display di destra vi appariranno dei numeri casuali che come sapete non hanno alcun significato, quindi impostate sulla tastiera il vostro primo codice, cioè 78, poi pigiate CONTROL — 0 per memorizzarlo.

4) Sulla cella successiva, cioè sulla 0101, dobbiamo inserire l'istruzione 48 quindi impostate questi due numeri sulla tastiera e pigiate ancora CONTROL — 0 per memorizzarli e passare alla riga successiva.

5) Nella riga 0102 scriveremo il numero 50 poi batteremo CONTROL — 0 per passare alla successiva.

6) Sulla riga 0103 scriveremo FF poi batteremo CONTROL — 0 ed a questo punto l'operazione inserimento programma in memoria può considerarsi conclusa.

Sui display rimarrà visualizzato il numero di riga 0104 seguito da due numeri casuali, per esempio:

infatti bisogna tener presente che noi abbiamo inserito delle istruzioni fino alla riga 0103 ed il computer, non sapendo che abbiamo già finito, attende che ne scriviamo una anche alla riga 0104, oppure che gli ordiniamo di fare qualcosa d'altro.

A questo punto se noi andiamo a rileggerci il contenuto delle righe di memoria dalla 0100 alla 0103 secondo i dettami dei paragrafi precedenti, sui display dovremo veder apparire:

Ovviamente se ci accorgessimo di aver commesso qualche errore, cioè di aver inserito per esempio nella riga 0101 il codice C8 anziché 48, potremo facilmente correg-

gere questo errore impostando sulla tastiera il nuovo codice e battendo quindi CONTROL — 0.

Una volta che avremo l'assoluta certezza di aver scritto in maniera corretta il programma, potremo finalmente passare alla fase più avvincente che consiste nell'esecuzione materiale del medesimo.

Innanzitutto dovremo scrivere il dato che ci interessa nel registro B; ammesso che questo dato sia per esempio il **C5** le operazioni che dovremo compiere saranno le seguenti:

1) Pigiare CONTROL — 2 per visualizzare sui display il contenuto del registro A.

2) Annotatevi su un foglio il numero casuale che troverete in questo registro per controllare in seguito se viene modificato poi pigiate CONTROL — 0 per passare al registro F.

3) Pigiare di nuovo CONTROL — 0 per passare al registro B.

4) Nel registro B dovete scrivere **C5** (volendo potreste inserire nel registro B anche altri dati per esempio 33-AA-3F-8D) quindi impostate tale codice sulla tastiera e pigiate CONTROL — 0 per passare al registro C.

5) Annotatevi su un foglio di carta il numero contenuto in questo registro.

6) Pigiare ancora CONTROL - 0 per passare al successivo registro D, annotatevi il numero in esso contenuto ed a questo punto avremo tutte le carte in regola per dire al nostro direttore, cioè alla CPU: « esegui il nostro programma ».

7) Come vi abbiamo detto il « direttore » non prende nessuna iniziativa autonoma, quindi se vogliamo effettivamente che esso esegua il nostro programma dovremo innanzitutto comunicargli dove può trovare sulla lavagna la prima istruzione da eseguire.

8) Per ottenere questo dovrete pigiare ancora CONTROL — 0 finché sui display non vi verrà visualizzato il contenuto del registro PC, cioè del Program Counter ed a questo punto dovrete impostare sulla tastiera il **numero della riga di memoria** in cui è contenuta la vostra prima istruzione, cioè **0100**. Quando sui display leggerete:



battete CONTROL — 0 per trasferire tale numero all'interno del registro.

9) Sui display vi apparirà il contenuto del registro A. A questo punto, per far eseguire al microcomputer il vostro programma, dovete semplicemente pigiare CONTROL - 4.

10) Immediatamente sui display vi apparirà:



(gli ultimi due numeri sono casuali)

e questo vi confermerà che il vostro programma è stato eseguito.

Infatti i **tre trattini orizzontali** sulla sinistra confermano

l'avvenuta esecuzione di un programma, i quattro numeri successivi indicano su quale riga della lavagna si è fermato il nostro direttore e i due display di destra ci indicano il contenuto di questa riga che a noi non interessa assolutamente in quanto non fa parte del programma.

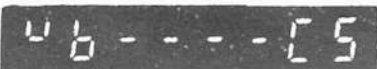
Ovviamente una volta ottenuto questo risultato sarete tutti curiosi di andare a vedere il contenuto dei vari registri per controllare se effettivamente il contenuto di B, cioè **C5**, è stato trasferito su A, su C e su D e per raggiungere questo scopo non dovrete fare altro che mettere in atto ciò che vi abbiamo spiegato nel precedente paragrafo: « Per leggere nei registri interni alla CPU ».

1) Pigiare CONTROL — 2 ed automaticamente vedrete comparire sui display:

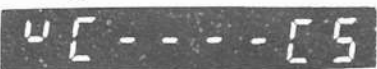


cioè su A c'è effettivamente scritto C5

2) Pigiare una prima volta CONTROL — 0 e poichè il contenuto di F non vi interessa, tornate a pigiarlo; vi apparirà:



3) Pigiare ancora CONTROL — 0 e vedrete:



4) Pigiare un'ultima volta e vedrete:



In altre parole, se non avete commesso errori ed avete seguito alla lettera le nostre istruzioni, ad esecuzione avvenuta su tutti e quattro i registri A-B-C-D deve essere scritto il codice C5 oppure quel codice diverso che voi avete inserito inizialmente nel registro B.

Potrete anche controllare il contenuto del Program Counter e qui senz'altro troverete:



cioè l'indirizzo della riga di memoria su cui il direttore si è fermato perchè nella riga prima ha trovato scritto STOP.

PROGRAMMA PASSO PER PASSO

Giunti a questo punto voi sapete già che cosa avviene nel nostro microcomputer pigiando i tasti CONTROL — 0, CONTROL — 1, CONTROL — 2, CONTROL — 4, però non sapete ancora perchè abbiamo volutamente saltato il CONTROL — 3 (è intuitivo che i comandi CONTROL — 5 e CONTROL — 6 si useranno solo quando collegheremo al microcomputer i registratori a nastro magnetico).

Ebbene questo comando, come avete già potuto appurare nella tabella iniziale, serve per far eseguire un pro-

gramma al microcomputer, istruzione per istruzione, anziché tutto di seguito.

Ovviamente un discorso di questo genere non è molto chiaro per chi non ha dimestichezza con i « computer » quindi per renderlo maggiormente comprensibile faremo ancora un esempio.

Supponiamo di avere un treno che partendo da Milano deve arrivare a Roma.

Pigiando CONTROL — 4 noi diciamo al treno di partire e quando questo è giunto a Roma il microcomputer ci dirà che è arrivato.

In altre parole noi sappiamo che l'operazione è stata eseguita però non sappiamo che strada ha seguito il treno, a quali stazioni si è fermato, se ha rispettato tutti i segnali lungo la linea oppure ne ha saltato qualcuno.

Generalmente queste informazioni potrebbero anche non interessarci infatti se il treno arriva a destinazione in perfetto orario e con tutte le carrozze previste è inutile andare ad indagare il percorso che ha seguito perché significa che le istruzioni che gli avevamo fornito erano esatte ed esso le ha eseguite alla perfezione.

Se però noi gli avessimo ordinato « parti da Milano e fermati a Bologna a caricare i passeggeri, riparti e fermati a Firenze sempre a caricare, riparti e fermati ad Orvieto a caricare poi prosegui fino a Roma e fermati », sapendo che il treno è arrivato a Roma e vedendo scendere pochi passeggeri, potrebbe assalirci il dubbio che questo si sia fermato solo a Bologna saltando Firenze e Orvieto, quindi per controllarne il percorso potremo utilizzare il CONTROL — 3.

Per mostrarvi in pratica come si utilizza questo comando ci rifaremo all'esempio precedente, cioè proveremo a far eseguire passo per passo il programma che trasferisce il contenuto del registro B, sui registri A-C-D.

Supponiamo di aver già scritto il programma nella memoria RAM e di aver già scritto il codice C5 sul registro B.

Per raggiungere lo scopo dovrete procedere come segue:

1) Pigiare CONTROL - 2 per accedere ai registri poi pigiate tante volte di seguito CONTROL - 0 fino ad arrivare al registro PC. Scrivete sul Program — Counter (registro PC) il numero della riga su cui è contenuta la prima istruzione del vostro programma, cioè 0100, poi pigiate CONTROL — 0 per trasferire questo numero all'interno del registro.

2) Pigiare CONTROL — 3 ed automaticamente vedrete comparire sui display:



Questo significa in pratica che il microcomputer ha già eseguito la prima istruzione (cioè 78 = trasferisci B su A) e si è fermato sulla seconda che come potrete facilmente controllare è 48 ed è contenuta nella riga 0101 della memoria RAM.

3) Adesso prima di far eseguire una seconda istruzione,

controlliamo se effettivamente il contenuto del registro B è stato trasferito su A e per far questo dovremo semplicemente pigiare CONTROL — 2.

Sui display ci apparirà:



a conferma che la prima istruzione è stata eseguita in modo corretto.

4) Per curiosità potreste anche controllare il contenuto di tutti gli altri registri.

Potrete anche verificare che sul registro PC è contenuto il numero 0101 cioè il numero della riga di memoria su cui è scritta la prossima istruzione del nostro programma.

5) Adesso che abbiamo verificato che il « treno » si è fermato alla prima stazione e ha caricato i passeggeri come richiesto, possiamo farlo ripartire per vedere se si ferma anche alla seconda e questo lo otterremo semplicemente pigiando CONTROL — 3.

6) Sui display ci apparirà immediatamente:



infatti il computer ha eseguito la seconda istruzione (cioè trasferisci B su C) e si è fermato alla terza, cioè 50, che è contenuta nella riga di memoria n. 0102.

7) Anche questa volta possiamo verificare il contenuto dei vari registri pigiando prima CONTROL — 2 poi CONTROL — 0 tante volte quanto necessita, in modo tale da verificare se il contenuto di B è stato trasferito anche su C oppure no.

8) Quando arriverete al registro C vedrete apparire sui display:



e questo vi confermerà che l'istruzione è stata eseguita correttamente.

Sul registro PC troverete ovviamente il numero 0102, cioè la riga di memoria su cui il microcomputer si è fermato.

9) Dopo aver verificato che il treno si è fermato anche alla seconda stazione ed ha caricato i passeggeri, possiamo farlo ripartire pigiando CONTROL — 3 ed automaticamente vedremo comparire sui display:



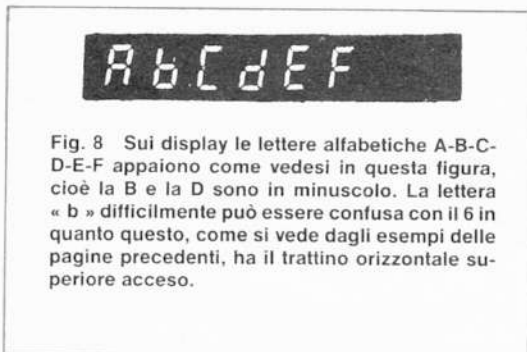
per confermarci che anche questa istruzione è stata eseguita e che resta da eseguire solo l'ultima, cioè lo STOP (FF) contenuto nella riga di memoria n. 0103.

10) Verificando ancora una volta il contenuto dei registri tramite CONTROL — 2 e CONTROL — 0 potremo a questo punto constatare che su A-B-C-D è presente il medesimo numero, cioè C5 e questo ci confermerà che il

nostro treno si è effettivamente fermato a tutte le stazioni indicate.

11) Se invece nello scrivere il programma noi ci fossimo sbagliati e nella cella 0102 avessimo scritto 60 al posto di 50, potremmo subito accorgerci dell'errore perché troveremo nel registro D un numero casuale ed in sua vece troveremo il registro H = C5 (infatti l'istruzione 60 trasferisce il contenuto di B su H).

Come vedete, sfruttando il comando CONTROL — 3 è molto facile svizzerare le pecche di un programma e porvi quindi rimedio.



QUALCHE PICCOLO ESERCIZIO

Vogliamo subito precisare che anche se un domani, quando ne sarete veramente « padroni », col vostro microcomputer potrete comandare una macchina utensile, eseguire delle operazioni aritmetiche estremamente complesse, tenere la contabilità di un magazzino ecc. ecc., è utopistico pensare che adesso, con una sola « lezione », si riesca ad imparare tutto questo. Cominceremo pertanto con esempi semplicissimi, facendo eseguire al microcomputer dei calcoli o delle operazioni così elementari che forse, eseguendoli a mente, richiederebbero un minor tempo, però sono proprio queste operazioni elementari che opportunamente combinate ci permetteranno un domani di eseguire correttamente programmi molto più complessi.

Possiamo anche precisarvi, e questo è importante per chi inizia, che pigiando a caso i tasti del microcomputer NON CAUSEREMO MAI NESSUN DANNO, così come non causeremo nessun danno al circuito scrivendo in memoria delle istruzioni errate: al massimo, nella peggiore delle ipotesi, vedremo bloccarsi tutti i comandi ed in tali circostanze, per ripristinare la normalità, non dovremo fare altro che pigiare il tasto di RESET in modo da restituire il controllo della situazione al MONITOR.

Poiché non abbiamo ancora le cassette, tutto ciò che scriverete all'interno della memoria RAM, togliendo alimentazione al circuito automaticamente si cancellerà, quindi se riuscirete a far « girare » un programma e vi

interessa che questo non vada completamente perduto, ricordatevi di trascrivere su un foglio di carta tutte le istruzioni riga per riga prima di spegnere il microcomputer, diversamente tutto il vostro lavoro se ne andrà automaticamente in fumo.

Se invece vi sarete scritti tale programma su un foglio di carta, quando accenderete di nuovo il computer potrete facilmente reinserirlo in memoria e sfruttarlo quindi per i vostri scopi.

Un'altra cosa da precisare è che attualmente sia i dati che inserirete in memoria, sia quelli che risulteranno dalle varie operazioni che farete compiere, saranno sempre in codice esadecimale, quindi non stupitevi se facendo eseguire al computer per esempio l'addizione $8 + 5$, il risultato non sarà 13 come vi aspettereste, bensì OD, infatti in esadecimale OD equivale a 13.

Analogamente se voi faceste eseguire al microcomputer l'operazione $18 + 18$ e vi aspettaste di veder comparire 36, provereste una grossa delusione perché il 18 che voi scrivete sulla tastiera è un numero esadecimale che in decimale equivale a 24, quindi la vera addizione che voi fate eseguire è $24 + 24$ non $18 + 18$ come credereste.

Un ultimo avvertimento che possiamo darvi, prima di passare alla pratica, è quello di non tentare per ora di modificare il contenuto del registro SP (cioè Stack Pointer) perché se non siete ancora degli esperti programmatori, inserendo su questo registro un numero non appropriato, potreste correre il rischio di veder cancellarsi i programmi che avete scritto nella memoria RAM.

A conoscenza di tutti questi particolari tenteremo ora di far eseguire al microcomputer una semplicissima addizione ($5 + 3$) scrivendo insieme a voi il relativo programma.

Per far questo useremo due registri interni alla CPU, cioè il registro A e il registro B.

Sul registro A scriveremo il numero 5; sul registro B scriveremo il numero 3 poi diremo al computer di sommare fra di loro questi due numeri e di scrivere il risultato ancora nel registro A.

In pratica le istruzioni che dobbiamo scrivere sono 4 e precisamente:

- 1) Carica sul registro A il n. 5
- 2) Carica sul registro B il n. 3
- 3) Addizione al contenuto di A il contenuto di B
- 4) STOP

Ricordandoci quanto abbiamo detto in precedenza, sappiamo già che la parola « carica » in mnemonico si scrive LD però LD da solo non significa niente infatti il « direttore », per eseguire questa istruzione, vuole sapere anche « dove » deve caricare e « cosa » deve caricare, quindi dopo LD dovremo aggiungere una A per specificare su quale registro vogliamo caricare il nostro numero, poi scrivergli sempre di seguito, interponendo una virgola, il numero vero e proprio, cioè 5.

In altre parole la prima istruzione in mnemonico si scrive:

LD A,5

tuttavia se voi andaste a ricercare questa istruzione nelle relative tabelle non riuscireste a trovarla perché essa viene indicata genericamente con

LD A,n

cioè « carica su A il numero n ».

Non vi sarà comunque difficile comprendere che noi non abbiamo fatto altro che **sostituire** in questo mnemonico **alla lettera n** che indica un numero generico, **il numero 5** che è quello che ci interessa.

La seconda istruzione del nostro programma è simile alla prima con la sola differenza che il registro su cui deve essere caricato il numero è il registro B e che il numero è un 3 invece che un 5, quindi possiamo subito scrivere il relativo mnemonico:

LD B,3

La terza istruzione inizia subito con la parola « addiziona » che noi sappiamo già si scrive **ADD** in mnemonico però **ADD** da solo anche qui non basta infatti il direttore, cioè la CPU, vuole sapere cosa deve aggiungere quindi seguendo la stessa falsariga dell'esempio precedente dovremo scrivere:

ADD A,B

dove A indica il registro su cui è contenuto il numero iniziale e B il registro su cui è contenuto il numero che gli vogliamo addizionare.

A questo punto abbiamo a disposizione tutti i mnemonici che ci servono quindi la prossima operazione che dovremo compiere sarà ricavarci, utilizzando le apposite tabelle riportate nel corso dell'articolo, i corrispondenti codici esadecimali da scrivere nella memoria RAM.

In queste tabelle noi troveremo che per « tradurre » in codice esadecimale l'istruzione **LD A,n** basta scrivere semplicemente **3E** però noi vogliamo dirgli di caricare il numero **5**, quindi *dopo* **3E** dovremo specificare anche **05**. Ne consegue che l'istruzione **LD A,5** dovrà essere scritta su due righe di memoria successive nel modo seguente:

3E

05

Lo stesso dicasi anche per l'istruzione **LD B,3** infatti se noi andiamo a guardare nelle tabelle troveremo **LD B,n = 06** però dato che dobbiamo specificare al calcolatore quale numero vogliamo caricare in B, dovremo scrivere:

06

03

La terza istruzione, cioè **ADD A,B**, si differenzia dalle precedenti per il fatto che è possibile trovarla tale e quale nella tabella n. 1 quindi per farla eseguire sarà sufficiente trascrivere il solo codice che troviamo di fianco ad essa, cioè:

80

Resta lo **STOP** finale per cui è sufficiente scrivere

FF

ed a questo punto il nostro programma è veramente pronto per essere caricato in memoria.

L'unico problema che dobbiamo ancora risolvere è quello di scegliere la riga di memoria in cui vogliamo scrivere la nostra prima istruzione ed ammesso di optare per la riga **0100**, le operazioni che dovremo compiere saranno le seguenti:

1) Pigiare il tasto di **RESET** per sbloccare il microcomputer nel caso in precedenza gli fosse stato dato un **HALT**.

2) Pigiare **CONTROL - 1** in modo da veder apparire sui display la scritta:

3) Impostate sulla tastiera esadecimale il numero **0100** quindi pigiate **CONTROL - 0** per vedere il contenuto di questa riga di memoria.

4) Sui display vi appariranno due numeri casuali, per esempio:

5) Impostate sulla tastiera il codice che volete inserire in questa riga, cioè **3E**, poi, controllate che sui display appaia la scritta:

pigiare **CONTROL - 0** in modo da trasferire questo codice all'interno della memoria RAM.

Sui display vi apparirà automaticamente il numero di riga successivo, cioè **0101**, seguito ancora da due numeri casuali, per esempio:

6) Impostate sulla tastiera il numero che volete inserire in questa riga, cioè **05**, e quando sui display vedrete:

Pigiare **CONTROL - 0** per trasferirlo alla memoria.

7) Sui display vi apparirà il numero di riga successivo, cioè **0102**, seguito ancora da due numeri casuali, per esempio:

8) Impostate sulla tastiera il terzo codice, cioè **06**, e quando sui display leggerete

pigiare **CONTROL - 0**

9) Sulla riga **0103** dobbiamo scrivere il codice **03**, quindi imposteremo tale codice sulla tastiera e non appena leggeremo:

igeremo ancora **CONTROL - 0**

10) Dopo lo **03** dovremo inserire il codice **80** che significa « Addiziona A + B » quindi imposteremo tale codice sulla tastiera e non appena leggeremo:

pigeremo **CONTROL - 0** per trasferirlo entro la memoria RAM.

11) Nella riga **0105** scriveremo **FF** e quando sui display leggeremo

batteremo **CONTROL - 0** per trasferirlo alla memoria.

A questo punto l'inserimento del programma in memoria può considerarsi terminato, tuttavia prima di farlo eseguire vogliamo ricontrollarlo per l'ultima volta.

Per far questo, come ormai saprete, pigeremo prima **CONTROL - 1** per riportare il computer in fase di « attesa », poi imposteremo sulla tastiera il codice **0100** corrispondente alla prima riga di memoria su cui abbiamo scritto le nostre istruzioni e successivamente pigeremo **CONTROL - 0** per vederne il contenuto.

Se non avete commesso errori, esplorando tutte le celle dalla 0100 alla 0105 dovreste trovare i seguenti codici:

riga 0100 =	3E
riga 0101 =	05
riga 0102 =	06
riga 0103 =	03
riga 0104 =	80
riga 0105 =	FF

A questo punto è giunta l'ora della verità, cioè il momento di verificare se il programma che abbiamo scritto funziona oppure no.

1) Pigiare **CONTROL - 2** poi pigiate tante volte di seguito **CONTROL - 0** finché non vi verrà mostrato il contenuto del registro **PC**, per esempio:

2) Scrivete su questo registro il numero **0100**, cioè la riga in cui è contenuta la prima istruzione del vostro programma, quindi pigiate **CONTROL - 0** per memorizzarlo.

3) Pigiare **CONTROL - 4** per far eseguire il programma ed automaticamente vedrete apparire il numero di riga **0106** seguito da due numeri casuali, per esempio:

Questo vi confermerà che il vostro programma è stato eseguito infatti se ora pigiate **CONTROL - 2** e andate a vedere il contenuto del registro A, sui display vi apparirà:

cioè proprio il risultato della nostra addizione ($5 + 3 = 8$).

Nota: in questo caso, essendo il risultato dell'addizione minore di 10, il numero esadecimale e il numero decimale, collimano tuttavia questa è solo un'eccezione che conferma la regola.

PROVATE ANCHE QUESTO

Nell'esempio precedente abbiamo volutamente addizionato fra di loro due numeri molto piccoli in modo tale da non confondervi troppo le idee con i codici esadecimali e decimali.

Adesso vogliamo complicarvi un po' la vita e ripetere lo stesso esempio con dei numeri un po' più alti per mostrarvi il modo in cui comunemente si procede.

Ricordiamo che nella memoria RAM noi abbiamo scritto le seguenti istruzioni.

Riga	Codice	Mnemonico	Spiegazione
0100	3E	LD A,n	Carica in A
0101	05	(n = 05)	il numero 05
0102	06	LD B,n	Carica in B
0103	03	(n = 03)	il numero 03
0104	80	ADD A,B	Addiziona A + B
0105	FF	STOP	Fermati

Questo programma, come già detto, ci permette di addizionare fra di loro i numeri 5 e 3 e di leggere alla fine il risultato nel registro A, cioè nell'Accumulatore.

Adesso noi vogliamo addizionare fra di loro, anziché i numeri 5 e 3, i numeri decimali 45 e 47 e per farlo sfrutteremo ancora lo stesso programma cambiando solo i numeri che abbiamo scritto nella riga 0101 (cioè il primo addendo) e nella riga 0103 (cioè il secondo addendo).

Per quanto detto in precedenza, su queste righe noi non potremo scrivere direttamente 45 e 47 che sono dei numeri « decimali », bensì dovremo prima preoccuparci di convertirli da decimale a esadecimale sfruttando a tale scopo le indicazioni fornite nell'articolo « il linguaggio del computer » presentato sulla rivista n. 67.

In pratica potete facilmente verificare che:

45 in esadecimale si scrive **2D**

47 in esadecimale si scrive **2F**

A questo punto potete senz'altro modificare il vostro programma, cioè porre nella riga 0101 il codice **2D** al posto di **05** e nella riga 0103 il codice **2F** al posto di **03** oppure, se lo preferite, riscrivere totalmente il vostro programma nel modo seguente:

Riga	Codice	Mnemonico	Spiegazione
0100 0101	3E 2D	LD A,n (n = 2D)	Carica in A il numero 2D
0102 0103	06 2F	LD B,n (n = 2F)	Carica in B il numero 2F
0104	80	ADD A,B	Addiziona A + B
0105	FF	STOP	Fermati

Non stiamo a ripetervi la procedura per inserire queste istruzioni all'interno della memoria RAM in quanto a tale argomento riteniamo di aver già dedicato anche troppo spazio nel corso di questo articolo quindi passeremo direttamente all'esecuzione vera e propria del programma.

Come al solito la prima operazione da compiere è quella di scrivere nel Program Counter (registro PC) il numero della riga in cui è contenuta la prima istruzione, cioè 0100 e per far questo dovremo:

1) Pigiare CONTROL - 2 per accedere ai registri interni alla CPU

2) Pigiare nove volte consecutive CONTROL - 0 per accedere al registro PC

3) Scrivere sulla tastiera esadecimale il numero 0100, poi battere ancora CONTROL - 0 per trasferire questo numero all'interno del registro.

Effettuata quest'ultima operazione, tutti all'interno del « ristorante cinese » saranno pronti per servirvi e per farlo aspetteranno solo che noi diciamo « VIA » pigiando CONTROL - 4. Immediatamente sui display comparirà il numero di riga 0106 seguito come al solito da due numeri casuali, cioè:

0106-62

e questo ci confermerà che il programma è stato eseguito per intero, infatti il registro PC contiene il numero di riga 0106 che è quello immediatamente successivo alla nostra ultima istruzione.

Ora se andate a vedere il contenuto del registro A pigiando CONTROL - 2 vedrete che questo è uguale a 5C,

4A-----5C

cioè non è il risultato che forse vi aspettavate da tale addizione in quanto trattasi, come già anticipato, di un numero esadecimale.

Per conoscere il « vero » risultato della vostra addizione dovrete quindi **convertire questo numero da esadecimale a decimale** ed automaticamente otterrete **92**, cioè proprio il risultato che cercavamo, infatti:

$$45 + 47 = 92$$

Precisiamo che con questo programma al massimo po-

tremo aggiungere fra di loro due numeri la cui somma non sia superiore a FF (cioè al numero decimale 255) perché è questo il numero massimo che può essere scritto nel registro A.

UN'ADDIZIONE DIRETTAMENTE IN DECIMALE

Tutto ciò che vi abbiamo detto di fare nel paragrafo precedente è piuttosto laborioso e poiché siamo certi che molti di voi preferirebbero ottenere subito il risultato dell'addizione in « decimale » senza dover operare le varie conversioni, vi insegneremo qui di seguito come sia possibile farlo.

Diremo subito che il problema è di facile risoluzione in quanto fra le 700 istruzioni circa che possono essere eseguite dallo Z80 ne esiste una (l'istruzione DAA) che effettua questa conversione in modo automatico.

In pratica, supponendo di voler eseguire la stessa addizione precedente (cioè 45 + 47), non dovremo far altro che scrivere nella riga 0101 direttamente il numero decimale 45 (anziché 2D come avevamo fatto in precedenza), nella riga 0103 il numero decimale 47 (anziché 2F) ed aggiungere quindi alla fine di tale programma il codice 27 che appunto corrisponde all'istruzione DAA.

Per meglio chiarirvi queste modifiche vi riportiamo comunque qui di seguito il programma tale e quale deve essere scritto nella memoria RAM.

Riga	Codice	Mnemonico	Spiegazione
0100 0101	3E 45	LD A,n n = 45	Carica in A il num. decim. 45
0102 0103	06 47	LD B,n n = 47	Carica in B il num. decim. 47
0104	80	ADD A,B	Addiziona A + B
0105	27	DAA	Converti in decimale
0106	FF	STOP	Fermati

Una volta scritto questo programma all'interno della memoria RAM, seguendo le indicazioni fornite nell'apposito paragrafo, dovremo inserire nel registro PC il numero di riga della prima istruzione, cioè 0100 agendo nel modo che ormai dovrebbe esservi familiare, cioè:

1) Pigiare CONTROL - 2 per accedere ai registri

2) Pigiare CONTROL - 0 finché non vedrete comparire sui display la sigla PC

3) Scrivete sulla tastiera 0100 poi battete CONTROL - 0

A questo punto potrete pigiare CONTROL - 4 per far eseguire il programma ed automaticamente sui display vedrete comparire il numero di riga 0107 seguito da due

numeri casuali a conferma che il programma stesso è stato eseguito.

Se ora pigiate CONTROL - 2 per verificare il contenuto del registro A vi troverete:



cioè proprio il risultato della vostra addizione direttamente in « decimale » (infatti $45 + 47 = 92$).

Precisiamo che con questo programma si possono sommare fra di loro due numeri che non diano come risultato un numero superiore a 99 perché è questo il massimo numero decimale che può essere contenuto nel registro A.

IL RISULTATO DIRETTAMENTE SUI DISPLAY

Con l'ultimo esercizio svolto abbiamo già fatto un passo da gigante in quanto ci siamo evitati di convertire prima in esadecimale i due numeri che volevamo addizionare fra di loro, poi di convertire da esadecimale a decimale il risultato contenuto nell'accumulatore.

Siamo certi però che qualcuno di voi vorrebbe avere ancora un qualcosa di più e precisamente desidererebbe che il risultato finale dell'operazione venisse visualizzato direttamente sui display, anziché dover ogni volta andarlo a cercare nel registro A. Ebbene ottenere questo è una cosa estremamente semplice ed alla portata di tutti; basta solo fare un piccolo ragionamento.

Abbiamo visto in precedenza che quando ha termine l'esecuzione di un qualsiasi programma, sui display compare sempre il numero di riga immediatamente successivo all'ultima istruzione eseguita e di fianco compare il numero contenuto in questa riga di memoria.

Per esempio il programma precedente finiva alla riga 0106 e quando noi lo abbiamo fatto eseguire, sui display è comparso il numero 0107 con a fianco il contenuto di tale riga che abbiamo detto era un numero casuale.

Se però noi, prima che il programma abbia termine, sapendo che l'istruzione di STOP è contenuta per esempio nella riga 0109, carichiamo il contenuto dell'accumulatore (cioè il risultato della nostra addizione) nella riga immediatamente successiva, vale a dire nella riga 010A, quando il computer si fermerà e ci mostrerà il contenuto di questa riga, automaticamente ci mostrerà il risultato dell'operazione che gli abbiamo fatto eseguire.

In altre parole, poiché conosciamo in partenza la riga di memoria che ci verrà visualizzata al termine dell'esecuzione del programma, noi non facciamo altro che scrivere in questa riga il numero che vogliamo ci venga mostrato ed automaticamente abbiamo raggiunto il nostro scopo.

Rifacendoci all'esempio precedente, le istruzioni da scrivere all'interno della memoria RAM per ottenere tutto questo sono le seguenti:

Riga	Codice	Mnemonico	Spiegazione
0100 0101	3E 45	LD A,n n = 45	Carica in A il numero 45
0102 0103	06 47	LD B,n n = 47	Carica in B il numero 47
0104	80	ADD A,B	Addiziona A + B
0105	27	DAA	Converti in decimale
0106 0107 0108	32 0A 01	LD (nn), A nn = 010A	Carica il contenuto di A nella riga 010A
0109	FF	STOP	Fermati

Come noterete l'istruzione LD (nn), A deve essere scritta su **3 righe di memoria successive**, non solo ma per scrivere 010A noi scriviamo **prima le due cifre sulla destra**, cioè 0A, **poi le due sulla sinistra**, cioè 01.

Per far eseguire questo programma inseriremo ancora nel Program Counter (registro PC) il numero 0100 seguendo le indicazioni fornite nei paragrafi precedenti, quindi pigeremo CONTROL - 4 ed automaticamente sui display vedremo comparire:



dove il 92 sulla destra rappresenta appunto il risultato della nostra operazione.

Ovviamente, avendo due cifre a disposizione, il numero massimo che potremo far comparire sarà anche in questo caso 99, perché se superassimo il 100 vedremo solo le due cifre sulla destra (per esempio se il risultato dell'addizione fosse 156, sui display ci apparirà solo 56).

PER CONCLUDERE

Sappiamo benissimo che quello che vi abbiamo insegnato a fare finora con il nostro microcomputer è ben poco rispetto alle possibilità effettive di tale circuito e sappiamo anche che molti di voi preferirebbero che gli si dicesse subito come si possa utilizzare il microcomputer per tenere la contabilità di un'azienda, per fare il carico e lo scarico di un magazzino oppure per controllare una macchina utensile.

Infatti chi non sarebbe contento, una volta acquistata una « macchina » di questa potenza, di sfruttarla subito al massimo delle prestazioni per ricavarne tutti i benefici possibili, però questo è pura fantascienza perché sappiamo benissimo che qualsiasi macchina si può sfruttare completamente solo quando la si conosce benissimo nei

minimi particolari e per raggiungere questo scopo ci vuole sempre molto tempo e molto esercizio.

Tanto per portare uno dei soliti esempi esplicativi, sarebbe come se un istruttore pilota per aerei a reazione, subito dopo aver insegnato al proprio allievo a cosa serve la « cloche », quale è la leva del gas e avergli detto che prima di atterrare è necessario far uscire il « carrello », gli ordinasse di prendere l'aereo e fare un bel volo rovesciato oppure un « giro della morte », cioè di eseguire quelle acrobazie che solo la « squadra tricolore », sottoponendosi ad un allenamento continuo, è in grado di compiere.

Certamente se l'istruttore adottasse questo metodo molti piloti, una volta in volo, preferirebbero abbandonare l'aereo scendendo con il proprio paracadute e lo stesso dicasi anche per i nostri lettori se anziché procedere con il metodo dei piccoli « passi », avessimo deciso di insegnarvi subito a calcolare per esempio l'arcotangente o il logaritmo di un numero.

È chiaro infatti che passando subito al « difficile », solo i più esperti possono riuscire a seguirci mentre tutti gli altri, cioè tutti quelli che veramente hanno voglia di imparare un qualcosa di nuovo, si perdono automaticamente per strada e poiché a noi non interessa insegnare a chi già sa bensì vogliamo svelare i « segreti » dell'elettronica anche a coloro che non hanno potuto seguire corsi a livello universitario, l'unica strada che potevamo scegliere era quella di iniziare proprio dalle cose più semplici, anche se queste per qualcuno sono cose « ovvie ».

D'altra parte, per rendersi conto che questo modo di procedere è giusto, basta pensare che molte riviste hanno iniziato già da tempo a parlare di microprocessori però avendolo fatto con un linguaggio non alla portata di tutti e soprattutto dando per scontata la conoscenza di certi termini inglesi di uso comune, quale risultato hanno ottenuto?

Hanno ottenuto che solo pochissimi sono riusciti a capire i loro discorsi infatti malgrado tutte queste pubblicazioni esiste ancora una stragrande maggioranza di lettori che non ha capito che cosa è un microcomputer e quali prestazioni se ne possono ricavare.

Noi al contrario, mentre gli altri parlavano, montavamo e collaudavamo questo e quell'altro microprocessore per scoprirne pregi e difetti ed oggi, con l'esperienza acquisita, siamo in grado non solo di offrirvi un progetto validissimo, superiore a tanti altri ed anche notevolmente più economico, bensì possiamo anche allacciare con voi un discorso « terra-terra », cioè una specie di conversazione fra amici in cui uno cerca di trasferire all'altro la propria esperienza corredando i vari discorsi di un'infinità di consigli pratici.

Solo in questo modo chi non sa può riuscire a comprendere, non certo con discorsi cattedratici stile professore universitario.

Quindi non abbiate fretta di arrivare alla fine bensì cercate di seguire attentamente quanto noi oggi vi diciamo e

quanto vi diremo in seguito, effettuando tutte le prove di volta in volta proposte, perchè trattandosi di un argomento molto complesso è sufficiente perdere una sola frase per non riuscire a comprendere più niente di ciò che si dirà in seguito.

Per chi ci ha richiesto quali saranno le prossime « espansioni » del progetto precisiamo inoltre che la prima scheda che presenteremo sarà quella che ci permetterà di collegare al microcomputer due registratori a nastro, vale a dire un'interfaccia molto sofisticata che a differenza di tante altre non risente dei disturbi, non ha problemi se il registratore è il più economico che esiste in commercio, non risente del livello di registrazione né di variazioni di velocità di trascinamento del nastro e questo si è potuto ottenere solo perfezionando passo per passo i nostri primi prototipi.

Passeremo successivamente a presentarvi la scheda necessaria per visualizzare sullo schermo del vostro TV casalingo i dati in caratteri alfanumerici e chi non vorrà utilizzare il proprio TV perchè la moglie non glielo permette, potrà disporre anche di un monitor da 12 pollici.

Insieme all'interfaccia video vi daremo anche la tastiera alfanumerica, simile cioè a quella delle macchine da scrivere, non solo ma vi daremo anche un interprete « BASIC » probabilmente bilingue, cioè in inglese e italiano, una novità questa assoluta.

Nota: per chi non ne fosse a conoscenza precisiamo qui che una volta inserito nel microcomputer l'interprete Basic, i programmi non si scriveranno più in codice esadecimale, un'operazione questa sempre piuttosto complessa, bensì scriveremo sulla tastiera le varie istruzioni proprio come se parlassimo in italiano con il microcomputer e questa è un'agevolazione veramente importante.

Capirete che preparare un interprete di questo genere, cioè che comprende le nostre parole e le traduce quindi in codice binario non è un lavoro che si fa in un sera, senza contare gli ostacoli dei diritti internazionali per poter utilizzare il Basic.

Questi problemi comunque sono già stati risolti, quindi possiamo fin d'ora assicurare il lettore che lo porteremo fino al « traguardo » senza farlo inciampare, cioè non solo gli daremo un aereo perfetto ma lo faremo anche diventare un « esperto pilota » in grado di eseguire qualsiasi acrobazia con la massima competenza.

SUL PROSSIMO NUMERO

Il lettore troverà lo schema e la realizzazione pratica dell'interfaccia necessaria per pilotare uno o due registratori a nastro più una scheda di espansione della memoria RAM.