

Quando alcuni mesi fa apparve una notizia sulla stampa americana di una scheda di espansione di memoria con 128K di RAM dinamica per l'Apple II, in grado di emulare un floppy disk, ci siamo incuriositi molto ed aspettavamo con ansia che qualche ditta italiana si mettesse in moto per importarla. Ora di queste schede in Italia ce ne sono varie ed esaminiamone da vicino una: la RAMEX 128 importata dall'Informatique di Aosta. La RAMEX 128 è un prodotto della Vergecourt Ltd, una ditta inglese che si è specializzata nella produzione di schede di espansione e periferiche per l'Apple II e l'ITT 2020, l'Apple costruito dalla ITT in Inghilterra su licenza della Apple Computer Inc. Oltre alla RAMEX 128, la Vergecourt produce una scheda di memoria da 16K chiamata RAMEX 16, che può essere usata al posto della scheda linguaggi (Apple Language Card) per il Pascal, il CP/M con la scheda Softcard oppure per aumentare la memoria usando il Visicalc. Questa scheda, ed anche la RAMEX 128, adotta una topologia circuitale radicalmente diversa sia da quella della Apple Language Card che da quella della maggior parte delle schede di espansione da 128K in commercio in Italia. Per montare la scheda Apple Language Card bisogna togliere un chip di memoria

RAMEX 128 PER APPLE II

Floppy disk allo stato solido

di Bo Arnklit

Quando un programma deve compiere frequenti accessi al disco si giunge spesso ad una situazione in cui l'esecuzione è lenta solo a causa dei continui coinvolgimenti della memoria di massa. Tempo fa, ci era venuta voglia di realizzare un banco di memoria RAM nella quale duplicare l'intero contenuto di un dischetto: in questo modo, sarebbe stato possibile eseguire il programma molto più velocemente, accedendo alla RAM anziché al disco; alla fine, il contenuto della RAM sarebbe stato ritrasferito sul disco per l'aggiornamento.

(un "4116") dalla piastra madre dell'Apple ed inserire lo spinotto del cavo piatto della scheda al posto della RAM. Sulla scheda Apple Language Card vi è poi un altro chip di memoria che sostituisce quello appena tolto dalla piastra madre. Questo strano collegamento serve a portare sulla scheda i segnali di refresh generati dal circuito video dell'Apple come spiegato nel riquadro.

Descrizione

Nelle schede RAMEX il generatore di refresh è ONBOARD, ed è quindi indipendente dal circuito di refresh dell'Apple. Inoltre è stato progettato per eseguire il minimo numero indispensabile di cicli di refresh e perciò le memorie si riscaldano molto di meno. In pratica diventano al massimo tiepide mentre quelle dell'Apple ed anche quelle della Apple Language Card diventano roventi, specialmente d'estate. L'altro vantaggio è che si elimina la necessità di fare quello strano collegamento col cavo piatto, togliendo una delle RAM della piastra madre dell'Apple, e si è liberi di inserire la scheda in uno qualsiasi degli SLOT.

I 128 K di memoria della RAMEX 128 sono organizzati come 8 banchi da 16 K, ognuno dei quali è in realtà composto da un banco da 8K e due blocchi da 4K. Gli 8K di memoria sono mappati da \$E000 a \$FFFF, mentre i due blocchi da 4K sono mappati da \$D000 a \$DFFF. Poiché tutti e due occupano la stessa mappa di memoria non è possibile selezionarli contemporaneamente, ma come vedremo non è un problema. Il motivo per il quale non si può mappare tutti i 16K da \$C000 a \$FFFF e che l'area di indirizzamento da \$C000 a \$CFFF è riservata all'input/output e gli 8 Slot dell'Apple.

La selezione dei banchi di memoria e l'abilitazione/disabilitazione della scrittura

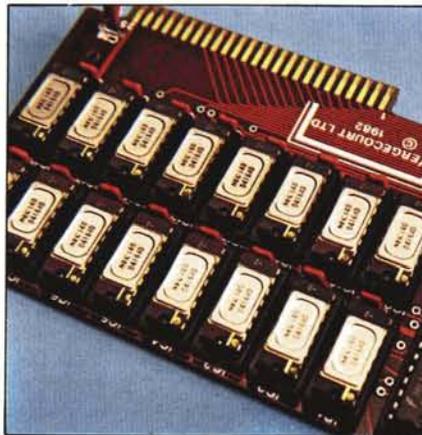
Costruttore:
Vergecourt Ltd., 17, Nobel Square, BASILDON, Essex. SS13 1LP.

Distributore per l'Italia:
Informatique, Avenue du Conseil des Commis, 14 - Aosta

Prezzo:
Super Ramex 128 K L. 699.000 + IVA

viene effettuata indirizzando alcune locazioni di memoria nella pagina \$C000 come mostra la figura 5.

Sulla scheda RAMEX 128 ci sono 3 LED che si accendono in codice binario per indicare quale degli otto banchi da 16K è stato selezionato. Questi LED sono utilissimi e quasi indispensabili durante lo sviluppo e la messa a punto di software in linguaggio macchina, per riuscire a capire "se sta facendo la cosa giusta". Gli altri tre LED presenti sulla scheda indicano (indipendentemente dal banco selezionato) se la scheda è stata selezionata per la lettura, la scrittura, quale dei due blocchi da 4K oppure se è deselezionata.



Le memorie usate nella scheda RAMEX sono le nuove memorie dinamiche da 64K x 1 bit. Come si può vedere bastano 16 integrati per fare 128K.

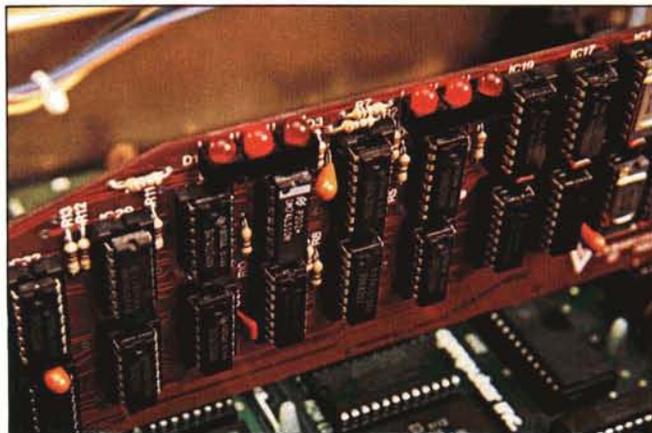
La RAMEX 128 viene fornita con un manuale in inglese ed un mini-floppy contenente il software necessario per usare la scheda come emulatore di disco. Come è noto la capacità dei dischi dell'Apple è di 140K byte, 35 tracce da 16 settori, ognuna da 256 byte, ma poiché il DOS occupa le prime tre tracce, rimangono 128K per i dati e la traccia 17 che contiene l'informazione relativa all'occupazione dei settori: il famoso VTOC (Volume Table Of Contents), oltre naturalmente alla directory o CATALOG. Poiché il DOS è già stato caricato in memoria sarebbe inutile ricaricarlo sulla scheda e perciò siamo nella fortunata situazione che 128K è precisamente quanto basta per emulare un disco.

Il software fornito insieme alla RAMEX 128 serve per trasformare il DOS presente sui dischi in un nuovo DOS che contiene un ulteriore RWTS speciale, per la gestione della RAMEX. Inoltre sono stati aggiunti dei nuovi comandi al DOS che ora si chiama SUPER HI-DOS. Questi nuovi comandi sono: #INIT, #MOUNT, #DUMP, #SAVE, #ATTACH, #PROTECT e #CLEAR. Per differenziarli dai normali comandi del DOS è necessario precederli con il "cancelletto" -#. Vediamoli una alla volta.

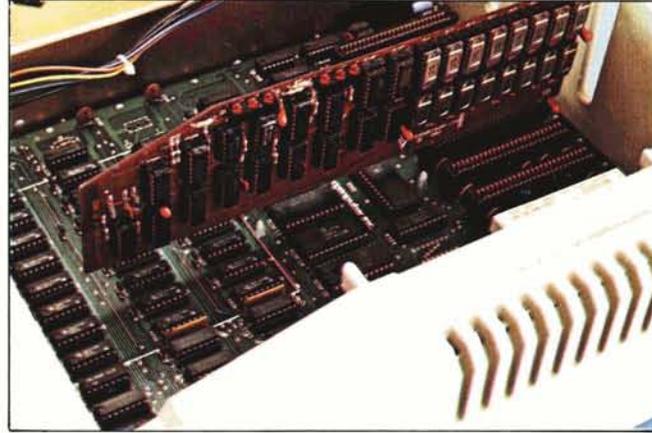
#INIT - Serve per inizializzare un floppy oppure la scheda RAMEX senza registrare il programma HELLO. Questo comando deve essere usato per INIT'ializzare la RAMEX prima di poterla usare come floppy disk.

#MOUNT - È il comando relativo al caricamento, o più propriamente, al trasferimento di tutti i dati (tranne il DOS, cioè le prime tre tracce) presenti sul disco, alla scheda RAMEX. Il tempo di trasferimento è di circa 90 secondi; forse non tra i più veloci, ma sempre più veloce di una copia tra due floppy.

#DUMP - Il comando #DUMP è l'esat-



I LED presenti sulla scheda sono utilissimi durante lo sviluppo del software, dando informazioni riguardanti il banco selezionato e lo stato della direzione di trasferimento dei dati.



La scheda RAMEX può essere inserita in uno qualsiasi degli SLOT dell'Apple II. Di solito si mette nello SLOT 4 o 5.

to contrario del #MOUNT. Effettua il trasferimento dalla scheda RAMEX al disco. Bisogna prestare particolare attenzione quando si usa questo comando perché una volta invocato, il contenuto del disco nel drive di destinazione verrà distrutto e sostituito dai dati nella RAMEX. Quindi attenzione a non lasciare il disco sbagliato nel drive.

#SAVE - Con questo comando è possibile salvare un segmento di un programma. Supponiamo di avere in macchina un programma di mille righe di cui vogliamo salvare la subroutine da riga 800 alla riga 900. Basta eseguire l'istruzione #SAVE SEGMENTO, 800, 900 e troveremo sul disco (o sulla RAMEX se abilitata) il segmento del programma nel file SEGMENTO. Può essere utile durante lo sviluppo di programmi permettendo appunto di salvarne pezzi per poi riprenderli e metterli insieme per fare un unico programma.

#ATTACH - Serve per caricare i segmenti di programma salvati con il comando #SAVE e aggiungerli ad un programma già esistente in memoria.

#PROTECT - È un comando che serve per WRITE - proteggere la RAMEX, in modo analogo al coprire la tacca sui dischi. Non ha nessun effetto se usato per un drive normale.

#CLEAR - È il contrario di PROTECT, serve cioè per abilitare per la scrittura la scheda RAMEX dopo che essa sia stata protetta con #PROTECT.

Uso come emulatore di disco

Dopo aver modificato il DOS sul nostro disco MASTER con il software fornito insieme alla scheda RAMEX siamo in grado innanzitutto di inizializzare la scheda eseguendo il comando #INIT,S5? se la scheda è in Slot 5, altrimenti basta cambiare il numero nel comando. (S4 per Slot4 etc). Questo può essere eseguito anche da programma con il solito sistema del PRINT per i comandi DOS:PRINT CHR\$(4);

#INIT,S5". Dopo questa operazione possiamo eseguire tutti i comandi normali del DOS oltre a quelli nuovi. Ad esempio possiamo copiare il disco in Slot 6, Drive I al "disco" in Slot 5 con il solito programma COPYA. La prima impressione è che i comandi brevi come il CATALOG oppure il LOAD di un programma di pochi settori, avvengono in modo praticamente istantaneo. Ciò è dovuto naturalmente al fatto che non si deve aspettare che il motore del drive avvii il disco ed arrivi ad una velocità costante. Entusiasmata da questa prima impressione abbiamo analizzato un po' più a fondo i tempi di caricamento di dati e programmi. Infatti succede che se si deve caricare un programma da circa 50 settori, (registrato su un disco vergine in modo che tutti i settori sono in fila uno dietro all'altro senza dover saltare da una traccia all'altra che altrimenti farebbe perdere tempo), il tempo di caricamento è aumentato solamente di circa due volte. Passiamo invece ad un'applicazione nella quale vengono eseguiti moltissimi accessi casuali al disco, come per esempio il riordino di dati direttamente sul disco. Nel numero 1 di MCmicrocomputer, nell'ambito della prova del CORVUS 5MB, abbiamo pubblicato un programma di Heapsort che opera direttamente sui file del disco. Riproponiamo questo programma in figura 1. Prima di farlo girare, però, bisogna creare un file con dei nomi casuali da riordinare; ci serviamo del programma della figura 2. Per avere un confronto diretto con i valori ottenuti nella prova del CORVUS facciamo un file con 100 nomi con una lunghezza di 8 caratteri. Il tempo di riordino con i floppy è di circa 12 minuti, con il CORVUS era di 4 minuti e mezzo ed ora con la RAMEX usata come emulatore di disco è di 3 minuti e mezzo. Quindi in questa particolare applicazione si ha un aumento rispetto al floppy di circa 3 volte e mezzo e rispetto al CORVUS di circa il 30 per cento. La causa di questa apparente lentezza è che l'unica cosa che cambia tra il DOS

normale ed il nuovo DOS è il RWTS, cioè la routine che va a leggere (o scrivere) su un dato settore di una traccia specifica. Ora invece di dare i comandi al braccio della testina di spostarsi sulla traccia desiderata ed aspettare che il settore giusto capiti sotto la testina, vengono eseguiti due conti per calcolare in quale banco da 16K ed a quale indirizzo esatto si trova il "settore", cioè i 256 byte relativi ad un settore fisico di un floppy. Dopo questo calcolo, i 256 byte vengono trasferiti ai buffer del DOS per l'analisi, proprio come il DOS normalmente. Ad esempio per caricare un file bisogna innanzitutto caricare il settore relativo al VTOC ed analizzarlo per sapere dov'è il CATALOG. Poi viene caricato il primo settore del CATALOG, analizzato per vedere se il nome del file esiste in questo primo settore del CATALOG. Se non c'è si ripete l'operazione fino a trovarlo, dopodiché si procede al caricamento dei settori relativi al file stesso, ogni volta trasferendoli prima nei file buffer del DOS e poi al suo indirizzo di destinazione. (Ad esempio un programma in Applesoft va caricato da \$803 in poi). A causa di tutti questi spostamenti tra la RAMEX ed i buffer del DOS la velocità di esecuzione è molto minore di quella che ci si aspetterebbe da un disco allo stato solido. Infatti, poiché la memoria della RAMEX (una volta che è stato selezionato il banco giusto) è memoria come il resto della memoria dell'Apple, basterebbe andare ad analizzare i "settori" direttamente dove stanno e trasferirli (quando li abbiamo trovati) direttamente al luogo di destinazione. Un DOS di questo genere sarebbe molto più veloce e speriamo che, in futuro, qualcuno lo produca.

Tuttavia la disponibilità di 128K di memoria non significa che devono essere necessariamente usati come emulatore di disco. Con un semplice programma in linguaggio macchina è possibile disporre di tutta questa memoria come memoria virtuale per creare ad esempio un Data Base in tempo reale. Il programma sorgente è

```

10 REM DISK HEAPSORT
20 REM COPYRIGHT 1981 BO ARNKLT
30 D$ = CHR$(4)
40 RF$ = D$ + "READFILE,R"
50 WF$ = D$ + "WRITEFILE,R"
60 CL$ = D$ + "CLOSE"
70 OP$ = D$ + "OPENFILE,L256"
80 HOME : VTAB 5: INPUT "NUMERO DI PAROLE ?":IMAX
90 PRINT OP$
100 N = IMAX:L = INT (N / 2 + 1):R = N
110 IF L > 1 THEN L = L - 1: PRINT RF$:L: INPUT R$: GOTO 130
120 PRINT RF$:R: INPUT R$: PRINT RF$:I: INPUT C$: PRINT WF$:R: PRINT
C$:R = R - 1: IF R = 1 THEN : PRINT WF$:I: PRINT R$: PRINT
CL$: GOTO 210
130 J = L
140 I = J:J = 2 * J: IF J < R THEN 170
150 IF J = R THEN 180
160 IF J > R THEN 200
170 PRINT RF$:J: INPUT B$: PRINT RF$:J + 1: INPUT A$: IF B$ < A
$ THEN J = J + 1
180 PRINT RF$:J: INPUT A$: IF R$ > = A$ THEN 200
190 PRINT RF$:J: INPUT C$: PRINT WF$:I: PRINT C$: GOTO 140
200 PRINT WF$:I: PRINT R$: GOTO 110
210 PRINT OP$: FOR J = 1 TO N: PRINT RF$:J: INPUT A$: PRINT A$:
NEXT : PRINT CL$

```

```

10 REM GENERATORE DI PAROLE RANDOM
20 REM COPYRIGHT 1981 BO ARNKLT
30 HOME : INPUT "NUMERO DI CARATTERI/PAROLA ?":C
40 INPUT "NUMERO DI PAROLE ?":IMAX
50 DIM NS(C): DIM A$(IMAX)
60 D$ = CHR$(4)
70 REM **** GENERAZIONE PAROLE **
80 PRINT D$:"OPENFILE,L256"
90 FOR J = 0 TO IMAX: FOR I = 0 TO C * .7 + 4 * C / 5 * RND (1
):A$ = A$ + CHR$( RND (1) * 26 + 65): NEXT
100 PRINT D$:"WRITEFILE,R":J
110 PRINT A$
120 A$ = ""
130 NEXT
140 PRINT D$:"CLOSE"

```

Figura 1 - A sinistra listato del programma di HEAPSORT per il riordino dei nomi contenuti nel file "FILE" usato nella prova di velocità di accesso.

Figura 2 - A destra programma per la generazione di nomi casuali con lunghezza variabile da usare con il programma di riordino di figura 1.

```

SOURCE FILE: RAMEX1
0000: 1 *****
0000: 2 :***
0000: 3 :*** Driver per Gestione Diretta
0000: 4 :*** della scheda RAMEX
0000: 5 :***
0000: 6 :*** (C) Copyright 1982
0000: 7 :*** Bo Arnklit MCmicrocomputer
0000: 8 :***
0000: 9 :*****
0000: 10 :
0000: 11 :
00E1: 12 CHRGET EGU $B1
0067: 13 FRMMUM EGU $DD67
DEC9: 14 SYNERR EGU $DEC9
E752: 15 GETADR EGU $E752
FDED: 16 COUT EGU $FDED
DFE3: 17 PTRGET EGU $DFE3
D995: 18 DATA EGU $D995
DAFB: 19 CRDO EGU $DAFB
0050: 20 LINUML EGU $50
0051: 21 LINUMH EGU $51
0000: 22 LEN EGU $00
0001: 23 RISUL EGU $01
0002: 24 BLOK EGU $02
0009: 25 ADDL EGU $09
000A: 26 ADDH EGU $0A
0006: 27 FLAG1 EGU $06
0007: 28 FLAG2 EGU $07
001A: 29 BEG1 EGU $1A
001B: 30 PTRIL EGU $1B
001C: 31 PTRIH EGU $1C
001D: 32 PTRZL EGU $1D
001E: 33 PTRZH EGU $1E
001F: 34 END1 EGU $1F
0003: 35 VARPNL EGU $B3
0000: 36 :
0000: 37 :
0000: 38 ; INIZIO PROGRAMMA
-----
NEXT OBJECT FILE NAME IS RAMEX1.OBJ0
0000: 39 ORG $9000
0000: 40 :
9000:20 B1 00 41 RAMEX JSR CHRGET ;Trova puntatori A$
9003:20 E3 DF 42 JSR PTRGET
9006:A0 00 43 LDY #0
9008:D1 03 44 LDA (VARPNL),Y
900A:85 00 45 STA LEN ;LEN=lunghezza A$
900C:CB 46 INY
900D:B1 03 47 LDA (VARPNL),Y
900F:05 18 48 STA PTRIL ;PTRIL,H indirizzo A$
9011:0B 49 INY
9012:B1 03 50 LDA (VARPNL),Y
9014:05 1C 51 STA PTRIH
9016:20 B1 00 52 NREC JSR CHRGET ;Numero record=NR=LINUMH,L
9019:20 57 DD 53 JSR FRMMUM
901C:20 52 E7 54 JSR GETADR
901F:A9 00 55 RECNUM LDA #0
9021:05 09 56 STA ADDL ;SET ADDL=0
9023:05 07 57 STA FLAG2 ;SET FLAG2=0
9025:A5 50 58 LDA LINUML
9027:29 7F 59 AND #$7F
9029:05 01 60 STA RISUL
902B:C9 60 61 CMP #$60
902D:90 00 62 BCC NOFLAG ;Record nei primi 12K
902F:A9 00 63 LDA #0 ;Record negli ultimi 4K
9031:05 07 64 STA FLAG2
9033:38 65 SEC
9034:A5 01 66 LDA RISUL
9036:19 60 67 SBC #$60
9038:85 01 68 STA RISUL
903A:A5 50 69 NOFLAG LDA LINUML
903C:2A 70 ROL A
903D:2A 71 ROL A
903E:29 01 72 AND #01
9040:05 02 73 STA BLOK
9042:A5 51 74 LDA LINUMH
9044:29 03 75 AND #$03 ;Solo records 0-1023
9046:0A 76 ASL A
9047:05 02 77 ORA BLOK
9049:05 02 78 STA BLOK
904B:1B 79 CLC
904C:46 01 80 LSR RISUL
904E:90 04 81 BCC STADDD
9050:A9 00 82 LDA #$80
9052:05 09 83 STA ADDL
9054:1B 84 STADDD CLC
9055:A5 01 85 LDA RISUL
9057:A9 00 86 ADC ADDL
9059:05 0A 87 STA ADDH ;ADDL,H=indirizzo record
905B:A5 51 88 LDA LINUMH ;Calcolo SLOT
905D:29 0C 89 AND #$0C
905F:1A 90 LSR A
9060:4A 91 LSR A
9061:AA 92 TAX
9062:80 29 91 LDA NSLOT,X
9065:05 10 94 STA PTRZL
9067:A9 C0 95 LDA #$C0
9069:05 1E 96 STA PTRZH
906B:20 B1 00 97 LOLMIT JSR CHRGET ;Carattere iniziale
906E:20 67 DD 98 JSR FRMMUM
9071:20 52 E7 99 JSR GETADR
9074:A5 50 100 LDA LINUML
9076:05 1A 101 STA BEG1
9078:20 B1 00 102 HILMIT JSR CHRGET ;Carattere finale
907B:80 9E 9B 112 LDA ERRMSG,X
907E:20 52 E7 104 JSR GETADR
9081:A5 50 105 LDA LINUML
9083:05 1F 106 STA END1
9085:C5 08 107 CMP LEN
9087:00 20 108 BEQ DPCODE
9089:90 28 109 BCC DPCODE
908B:A2 00 110 ERRBUF LDX #0
908D:20 FB DA 111 JSR CRDO ;Output Carr.Return
908F:80 9E 9B 112 LDA ERRMSG,X
9093:20 ED FD 113 JSR COUT
9094:EB 114 INX
9097:C9 00 115 CMP #$8D ;Fine messaggio?
9099:D0 F5 116 BNE PRMSG
909B:4C C9 DE 117 JSR SYNERR
909E:C2 D5 C6 118 ERRMSG ASC "BUFFER STRING TOO SHORT"
90A1:C6 C5 D2
90A4:A0 D3 D4
90A7:02 C9 CE
90AA:C7 A0 D4
90AD:CF CF A8
90B0:D3 CB CF
90B3:D2 D4
90B5:80 119 DFB $8D
90B6:1B 120 DFCODE CLC
90B7:A5 09 121 LDA ADDL
90B9:65 1A 122 ADC BEG1
90BB:05 09 123 STA ADDL
90BD:20 B1 00 124 JSR CHRGET
90BF:C9 52 125 CMP #$52 ;R-READ
90C2:F0 07 126 BEQ READ
90C4:C9 57 127 CMP #$57 ;W-WRITE
90C6:F0 07 128 BEQ WRITE
90C8:4C C9 DE 129 JSR SYNERR
90CB:A9 00 130 READ LDA #0
90CD:F0 02 131 BEQ STORE
90CF:A9 01 132 WRITE LDA #1 ;FLAG1=1 - WRITE
90D1:05 06 133 STORE STA FLAG1 ;FLAG1=0 - READ
90D2:00 F6 90 134 JSR BSWITC ;Bank Switching
90D6:A5 06 135 MOVE LDA FLAG1
90D8:D0 0E 136 BNE OUTPUT
90DA:A0 00 137 INPUT LDA #0 ;Trasf. RAMEX->MEM
90DC:B1 09 138 LD1 LDA (ADDL),Y
90DE:91 18 139 STA (PTRIL),Y
90E0:CB 140 INY
90E1:C4 1F 141 CPY END1
90E3:D0 F7 142 BNE LD1
90E5:4C 19 91 143 JMP FINE
90E8:A0 00 144 OUTPUT LDY #0 ;Trasf. MEM->RAMEX
90EA:81 18 145 LO2 LDA (PTRIL),Y
90EC:91 09 146 STA (ADDL),Y
90EE:CB 147 INY
90F0:1C 1F 148 CPY END1
90F1:D0 F7 149 BNE LD2
90F3:4C 19 91 150 JMP FINE
90F6:A6 02 151 BSWITC LDX BLOK
90F8:BC 21 91 152 LDY TABLE,X
90FB:81 10 153 LDA (PTRZL),Y ;READ $C0X4,5,6,7,C,D,E,F SWITCH BANK
90FD:A5 06 154 LDA FLAG1
90FF:D0 08 155 BNE RAMH
9101:A0 00 156 RAMH LDY #0
9103:A5 07 157 LDA FLAG2
9105:F0 02 158 BEQ SW1
9107:A0 08 159 LDY #8
9109:B1 10 160 SW1 LDA (PTRZL),Y ;READ C0X0,8-READ ENABLE,WRITE PROT
910B:60 161 RTS
910C:A0 03 162 RAMH LDY #3
910E:A5 07 163 LDA FLAG2
9110:F0 02 164 BEQ SW2
9112:A0 08 165 LDY #$8
9114:B1 10 166 SW2 LDA (PTRZL),Y ;DUE READ C0X3,8-WRITE ENABLE
9116:81 10 167 LDA (PTRZL),Y
9118:A0 168 RTS
9119:A0 02 169 FINE LDY #2 ;WRITE PROTECT RAMEX
911B:81 10 170 LDA (PTRZL),Y
911D:20 95 D9 171 LDA DATA ;Ritorno all'Applesoft
9120:A0 172 RTS
9121:04 05 06 173 TABLE DFB $4,$5,$6,$7,$C,$D,$E,$F
9124:07 0C 0D
9127:0E 0F
9129:D0 C0 B0 174 NSLOT DFB $00,$C0,$80,$A0 ;SLOT 5,4,3,2
912C:A0

```

Figura 3 - Programma sorgente scritto con l'Assembler del TOOLKIT. Si tratta di una routine che gestisce direttamente la scheda RAMEX come espansione di memoria.

riportato nella figura 3. È scritto usando l'assembler del TOOLKIT, che dovrebbe essere abbastanza diffuso tra i lettori. Per chi non possiede un assembler o per chi vuole solamente inserire il programma senza modificarlo può inserire i dati riportati nella figura 4. Il programma è stato assemblato per girare alla locazione \$9000 e perciò vanno inseriti come segue:

CALL-151 <Ret>

* 9000: 20 B1 00 20 E3 DF A0 00 ... etc.

Alla fine si può salvare su disco con il nome RAMEX:

BSAVE RAMEX,A\$ 9000, L\$ 12 D
<Ret>

Questa routine gestisce la RAMEX co-

me 1024 record da 128 byte, e l'interfaccia con l'Applesoft è particolarmente facile. Supponiamo di avere una stringa A\$ di lunghezza compresa tra 1 e 128 caratteri e di volerla immagazzinare nel record numero 823. Innanzitutto dobbiamo caricare la routine in linguaggio macchina che abbiamo appena preparato ed inserire la RAMEX nello slot 5. Se abbiamo più di una scheda RAMEX la seconda va nello Slot 4 e così via fino a 4 schede per un totale quindi di 4096 record da 128 byte. A questo punto facciamo semplicemente un CALL come segue:

CALL 36864; A\$,823,0,LEN (A\$), W
dove chiaramente 36864 è l'equivalente di

\$9000, A\$ è la nostra stringa, 823 è il numero del record che va da 0 a 1023 per la prima scheda, da 1024 fino a 2047 per la seconda, etc. fino a 4095 per la quarta. Lo zero e "LEN(A\$)" indicano rispettivamente la posizione all'interno del record (tra 0 e 128) ed il numero di caratteri da trasferire.

Infine la "W" sta per WRITE cioè trasferimento dalla memoria centrale dell'Apple, nella quale si trova la stringa A\$, alla scheda RAMEX. Per la lettura di un record dobbiamo semplicemente sostituire la "W" con una "R" (READ). Prima di effettuare un'operazione di lettura bisogna "dimensionare" la stringa di destinazione (nel nostro caso A\$). Supponiamo di aver

```

9000- 20 B1 00 20 E3 DF A0 00
9008- B1 83 85 00 C8 B1 83 85
9010- 1B C8 B1 83 85 1C 20 B1
9018- 00 20 67 DD 20 52 E7 A9
9020- 00 85 09 85 07 A5 50 29
9028- 7F 85 01 C9 60 90 0B A9
9030- 08 85 07 38 A5 01 E9 60
9038- 85 01 A5 50 2A 2A 29 01
9040- 85 02 A5 51 29 03 0A 05
9048- 02 85 02 18 46 01 90 04
9050- A9 80 85 09 18 A5 01 69
9058- D0 85 0A A5 51 29 0C 4A
9060- 4A AA BD 29 91 85 1D A9
9068- C0 85 1E 20 B1 00 20 67
9070- DD 20 52 E7 A5 50 85 1A
9078- 20 B1 00 20 67 DD 20 52
9080- E7 A5 50 85 1F C5 00 F0
9088- 2D 90 2B A2 00 20 FB DA
9090- BD 9E 90 20 ED FD EB C9
9098- 8D D0 F5 4C C9 DE C2 D5
90A0- C6 C6 C5 D2 A0 D3 D4 D2
90A8- C9 CE C7 A0 D4 CF CF A0
90B0- D3 C8 CF D2 D4 8D 18 A5
90B8- 09 65 1A 85 09 20 B1 00
90C0- C9 52 F0 07 C9 57 F0 07
90C8- 4C C9 DE A9 00 F0 02 A9
90D0- 01 85 06 20 F6 90 A5 06
90D8- D0 0E A0 00 B1 09 91 1B
90E0- C8 C4 1F D0 F7 4C 19 91
90E8- A0 00 B1 1B 91 09 C8 C4
90F0- 1F D0 F7 4C 19 91 A6 02
90F8- BC 21 91 B1 1D A5 06 D0
9100- 0B A0 00 A5 07 F0 02 A0
9108- 0B B1 1D 60 A0 03 A9 07
9110- F0 02 A0 0B B1 1D B1 1D
9118- 60 A0 02 B1 1D 20 95 D9
9120- 60 04 05 06 07 0C 0D 0E
9128- 0F D0 C0 B0 A0
*
```

Figura 4 - Codice oggetto del programma della figura 3, assemblato a \$9000.

CONTROL ADDRESSES	FUNCTION SELECTED	4K Bank SELECTED
HEX \$C080	Ramex Read and	Bank 2
DECIMAL -16256	Ramex Write Protect	
HEX \$C081	On Board Rom Read	Bank 2
DECIMAL -16255		
HEX \$C082	On Board Rom Read and	Bank 2
DECIMAL -16254	Ramex Write protect	
HEX \$C083	Ramex Read	Bank 2
DECIMAL -16253		
HEX \$C088	Ramex Read and	Bank 1
DECIMAL -16248	Ramex Write protect	
HEX \$C089	On Board Rom Read	Bank 1
DECIMAL -16247		
HEX \$C08A	On Board Rom Read	Bank 1
DECIMAL -16246	and Ramex Write protect	
HEX \$C08B	Ramex Read	Bank 1
DECIMAL -16245		

"On Board Rom" si riferisce alla Rom (Applesoft) presente sulla scheda madre.

* * *

Se i seguenti indirizzi sono selezionati due volte consecutivamente allora il Ramex è anche "Write enabled" come segue.

CONTROL ADDRESSES	FUNCTION SELECTED	4K Bank SELECTED
HEX \$C081	Write enables Ramex	Bank 2
DECIMAL -16255		
HEX \$C083	Write enables Ramex	Bank 2
DECIMAL -16253		
HEX \$C089	Write enables Ramex	Bank 1
DECIMAL -16247		
HEX \$C08B	Write enables Ramex	Bank 1
DECIMAL -16245		

Figura 5 - Tabella degli indirizzi che effettuano la selezione dei vari banchi di memoria e la scelta tra lettura o scrittura di dati.

Memorie dinamiche e ciclo di Refresh

Le memorie RAM usate nell'Apple II sono del tipo dinamico; vale a dire che il contenuto di una cella di memoria non viene immagazzinato per sempre, ma verrebbe perso dopo un breve intervallo, dopo un'operazione di scrittura o di lettura. È quindi necessario leggere tutte le locazioni di memoria periodicamente con un intervallo che non superi i 2 millisecondi. Questa operazione è chiamata Refresh Cycle (o ciclo di Refresh). Nell'Apple questo ciclo è affidato (molto astutamente) al circuito video che scansiona la memoria. Ricordiamo che il video dell'Apple II è memory mapped, vale a dire che ogni carattere dello schermo ha il suo preciso posto nella memoria RAM riservata per il video display ed indirizzabile direttamente dal microprocessore. Infatti l'area di memoria da \$400 a \$7FF (pari ad 1 K di memoria) viene utilizzata come mappa di memoria per la pagina 1 del testo. Un circuito elettronico provvede poi alla scansione di quest'area di memoria, leggendo ogni byte, passandolo attraverso il generatore di caratteri (che per molti di voi sarà ormai stato sostituito con l'Apple-Minus), mischiato con i segnali di sincronismo ed inviato al video. Un ciclo completo (lettura di tutti i caratteri e visualizzazione di un'intera schermata) avviene 50 volte al secondo. La lettura dei caratteri da parte del circuito avviene durante la fase 1 del Clock del sistema, quando il microprocessore è impegnato nella decodifica delle istruzioni e quindi non indirizza nessuna locazione di memoria. In questo modo il ciclo di visualizzazione dei caratteri sul video è completamente "trasparente"; vale a dire che non rallenta minimamente il microprocessore. Le memorie RAM dell'Apple sono del tipo 4116, 16K x 1 bit, e perciò ciascuna fila di 8 chip ci dà 16K byte. In realtà all'interno di questi chip i 16K bit sono organizzati in una matrice da 128 x 128 bit. Per garantire l'integrità dei dati basta indirizzare ciascuna delle 128 righe della matrice ogni 2 millisecondi. In un secondo sono dunque necessari 64000 accessi alla memoria. Nell'Apple il ciclo di refresh è affidato al circuito video che, come abbiamo visto, deve indirizzare almeno 1K di memoria per ogni quadro del video, e quindi si possono utilizzare le 7 linee di indirizzo meno significative per indirizzare le 128 righe delle RAM dinamiche. Come abbiamo visto, però, quest'indirizzamento occorre ogni volta che il Clock passa per la fase 1, cioè 1.023.000 volte al secondo. Il ciclo di refresh è dunque circa 16 volte più frequente del necessario. La soluzione adottata dalla Apple è molto elegante, perché si risparmia tutta la circuiteria altrimenti necessaria per la generazione dei cicli di refresh, ma presenta uno svantaggio: poiché infatti le memorie dinamiche dissipano più potenza quando sono attive (indirizzate), rispetto a quando sono in stand-by, di un fattore di circa 15, è chiaro che meno si indirizza il chip, anche per il refresh, e meno questo si riscalda.

organizzato il nostro record in modo di avere il nome, cognome, indirizzo, CAP e città tutti di lunghezze fisse, con il CAP situato per esempio dal carattere 80 fino al carattere 84 (incluso). Per trasferire in A\$ solo i caratteri relativi al CAP bisogna innanzitutto creare una stringa A\$ che sia lunga 5 caratteri, facendo per esempio: FOR I=1 TO 5 : A\$=A\$+"X": NEXT oppure direttamente (essendo di soli 5 caratteri):

```
A$="XXXXX"
```

Poi facciamo il nostro CALL:
CALL 36864; A\$,823,80,5,R

Per caricare tutti i CAP dell'intera scheda in un array chiamato per esempio R\$(I) possiamo fare:

```
DIM R$(1023)
```

```
A$="XXXXX"
```

```
FOR I=0 TO 1023
```

```
CALL 36864; A$,I,80,5,R
```

```
R$(I)=A$
```

```
NEXT
```

A titolo di esempio della velocità di esecuzione questa operazione richiede circa 7 secondi per il trasferimento di tutti i 1024 "CAP". Una operazione simile eseguita su un file RANDOM di un floppy dove prima di tutto bisogna leggere il record, poi scomporlo in sottostringhe (con MID\$(A\$,80,5)) richiederebbe più di 10 minuti per lo stesso numero di record. Perciò l'aumento di velocità è di circa 100 volte; non i 3 o 4 della RAMEX usata come emulatore.

La routine presentata è ovviamente un po' limitata per il fatto di usare dei record con lunghezza fissa di 128 byte; tuttavia lo abbiamo usato con delle leggere modifiche come cuore di un Data Base di altissima velocità (appunto in tempo reale) nella quale è stato adoperato tra l'altro l'Heap-sort in linguaggio macchina pubblicato sul numero 3 di MCmicrocomputer. Forse in futuro ne riparleremo.

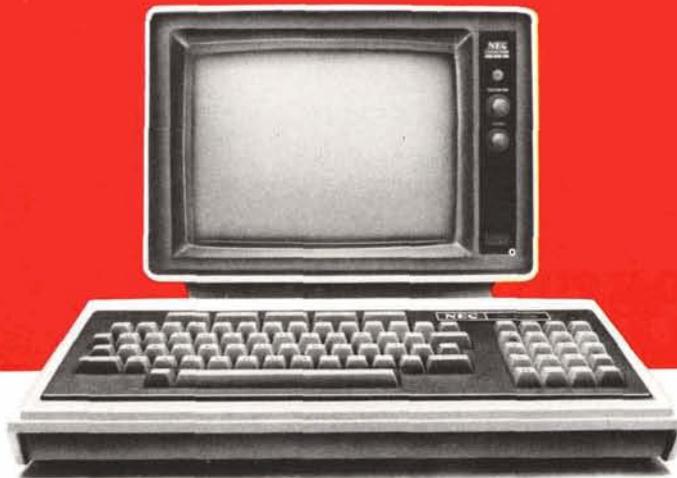
Un'altra applicazione interessante della scheda RAMEX 128, alla quale accenniamo solamente, è come espansione di memoria per il programma VISICALC. È infatti disponibile uno speciale disco di BOOT per caricare il VISICALC con la RAMEX espandendolo fino a 136K!

Conclusioni

La Ramex 128 è senza dubbio un oggetto di impostazione abbastanza "particolare", molto utile a chi ha la necessità di trattare un grosso numero di dati immagazzinati, in uno o più minifloppy accedendo frequentemente (e rapidamente) a ciascuno di essi. Usata con un po' di "fantasia", è in grado di dare risultati a dir poco sorprendenti: abbiamo visto persone rimanere letteralmente a bocca aperta di fronte al nostro "real-time data base" che usa tre schede. Il tutto ad un prezzo che, considerando le prestazioni ed a patto, ripetiamo, di sfruttare adeguatamente le possibilità della scheda, appare ragionevolmente contenuto.

personal computer

NEC



NIPPON ELECTRIC CO., Ltd.

LEADER IN JAPAN

SERIE

PC-8000

hal
computers

HAL COMPUTERS s.r.l. - Direzione: Via Pier Capponi 11 - 20145 MILANO
Tel. 02/4980783 - 4696037 - Telex 331422 FESTUD I
DISTRIBUTORE PER L'ITALIA

SOFTWARE UNIVERS

SI CONFRONTANO IN UN MATCH APPASSIONANTE
L'HP-86 e L'OLIVETTI M20 ST.

**PRONTA
CONSEGNA
LEASING
IMMEDIATO**

IL NOSTRO PUNTO DI FORZA IL SOFTWARE

Unis INPUT

La prima fase del programma consente in maniera incredibilmente semplice e colloquiale l'input dei dati generali della struttura, la correzione e l'archiviazione automatica dei dati stessi.

CALCOLO

La fase di calcolo vera e propria esegue a prova di normativa:

- l'analisi dei carichi (ripartiti concentrati coppie, dilatazioni termiche, ecc.), alterando i carichi accidentali e le forze sismiche, vengono sviluppate fino a 15 condizioni di carico diverse;
- l'analisi sismica calcolo baricentro delle masse e delle rigidità, eccentricità, calcolo delle forze sismiche e ripartizione delle stesse a livello di tutti gli elementi della struttura (in particolare setti, corpi soia, elementi fuori squadra, ecc.);
- analisi torsionale: calcolo del centro di torsione e distribuzione degli effetti torsionali fra i vari elementi della struttura;
- risoluzione dei vari telai longitudinali e trasversali e calcolo delle caratteristiche di sollecitazione nelle varie sezioni;
- progetto-verifica delle varie sezioni di ogni elemento e determinazione delle aree di armatura e/o del numero dei tendini nelle sezioni stesse. Durante le due fasi precedenti il programma va a crearsi automaticamente tutti i files di lavoro ossia archivia tutti i dati necessari per i calcoli, per le stampe e per i disegni successivi.

STAMPE

Automaticamente vengono stampate le seguenti parti:

- relazione di calcolo;
- dati geometrici della struttura;
- carichi agenti;
- baricentri delle masse e delle rigidità e forze sismiche (così come richiesto dalle recenti norme sugli edifici in zona sismica);
- momenti ai nodi per i vari telai e per tutte le condizioni di carico richieste;
- caratteristiche delle sollecitazioni nelle varie sezioni di tutti gli elementi;
- progetto-verifica delle sezioni stesse con indicazione delle aree di armatura e tutti i riferimenti di normativa (armature minime ecc.).

ELABORATI GRAFICI

Sequenzialmente il programma esegue sul plotter in modo completamente automatico tutti quei disegni esecutivi (su carta lucida e china) necessari al completamento del progetto quali i diagrammi del momento del taglio e dello sforzo normale, i disegni delle deformate, il disegno delle piante e delle sezioni ed infine esegue il disegno delle carpenterie e delle armature dei vari elementi (travi, pilastri, travi di fondazione, plinti, ecc.).

COMPUTI

Il programma esegue infine i computi metrici del c.a. e dell'acciaio occorrenti per la realizzazione della struttura.

Stute

Analisi generale di strutture spaziali generiche in fase statica e in fase dinamica (primi N modi di vibrazione), il metodo di calcolo è agli elementi finiti. La geometria della struttura prevede:

- elementi generici (aste comunque inclinate con variazione di inerzia lineare e/o parabolica);
 - qualsiasi tipo di vincolazione interna ed esterna (carrelli e cerniere interni, cedimenti, variazioni termiche, distorsioni, ecc.).
- Linee d'influenza per azioni mobili. Come output il programma, indipendentemente dalle caratteristiche del materiale di cui la struttura è costituita, dà tutte le caratteristiche di sollecitazione e di formazione agli estremi di ogni asta (e quindi di qualsiasi sezione dell'asta).

Il programma è particolarmente efficace sia per strutture omogenee (cemento armato o acciaio ecc.) sia per strutture miste. Inoltre consente lo studio dei «meccanismi e dei modi rigidi», problema notevolmente complesso nel campo dell'ingegneria meccanica.

Automaticamente il programma, dalla fase di input in poi fornisce una serie di stampe, di tabelle e di grafici, ovvero una completa relazione di calcolo. Il manuale per l'uso del programma costituisce un vero e proprio trattato di Scienza delle Costruzioni.

Ponti MASSONET

Il programma calcola la ripartizione trasversale dei carichi secondo il metodo di Massonet, il quale tiene conto della deformabilità dei travi in contrapposizione al metodo di Courbon che li considera infinitamente rigidi. Il programma chiede le caratteristiche di inerzia delle travi e dei traveri, dopodiché assume le ipotesi del metodo calcola i coefficienti di ripartizione sulle varie travi, quando il carico unitario viene posto su una generica trave.

GRIGLIATI PIANI

Il metodo di risoluzione a grigliati cioè di strutture aventi la caratteristica di essere caricate normalmente al proprio piano, è quello adottato nei computers di grandi capacità. Siamo riusciti ad inserire questo metodo sull'HP grazie alle notevoli caratteristiche del suo sistema operativo!

Il metodo consiste nel creare una matrice di rigidità generale della struttura in cui si tiene conto della deformabilità flessionale e torsionale delle membrature componenti. Questo metodo di analisi matriciale dà al progettista la possibilità di risolvere numerosi casi di strutture (piastra, solette, grigliati), vincolate in modo generico (appoggi, incastri, cerniere, appoggi elastici) e caricate in modo qualsiasi e in qualsiasi zona della struttura stessa. L'output consiste nelle caratteristiche di deformazione alle estremità di ogni membratura.

TRAVI IN C.A.P.

È la trattazione completa del progetto verifica di strutture in C.A.P. quali travi, solette, cassette nel caso di cavi pretesi e/o postesi con andamento non simmetrico soggette a carichi generici nelle varie fasi della precompressione. Le sezioni sono a geometria generica e variabile lungo l'asse della struttura.

CALCOLO DELLE SPALLE

Questo programma permette il calcolo di strutture di fondazione quali spalle, muri di sostegno ecc. in c.a. e a gravità con particolare riferimento alle recenti norme antisismiche. La geometria della struttura è di tipo generico e la fondazione può essere sia diretta che su pali eventualmente inclinati. È possibile tener conto di qualsiasi tipo di carico sia verticale che orizzontale indipendentemente dalla forza sismica che viene calcolata auto-



HP-86

OLIVETTI M20 ST.



maticamente.

Questo programma rappresenta veramente tutto quanto necessario nella progettazione di strutture di questo tipo anche per il fatto che il calcolatore, al fine di ottimizzare il procedimento di calcolo visualizza tutta una serie di risultati intermedi; esegue tutte le verifiche di stabilità, il progetto verifica delle armature nelle varie sezioni e infine stampa la relazione di calcolo definitiva.

General 86

CALCOLO AGLI STATI LIMITI

Per le sezioni in c.a. secondo la normativa del DM 26 marzo 1980, risolve il problema della flessione semplice e della pressotensoflessione sia in fase di progetto che in fase di verifica. Determina il campo di rottura, il momento ultimo, le deformazioni unitarie dei materiali e il coefficiente di sicurezza per le varie sezioni sia rette che circolari, sia piene che cave.

SEZIONI CIRCOLARI

Progetto e verifica di sezioni circolari inflesse e pressotensoflesse piene o cave in c.a. con il metodo non armatura disposta simmetricamente su una corona di raggio R con eventuale

le armature suppletiva posta in trazione e compressione.

SOLAIO CONTINUO: verifica dei momenti agli appoggi e nelle campate e disegno dei diagrammi su stampante.

TRAVI CONTINUE: calcolo delle travi continue comunque vincolate e sottoposte a qualunque tipo di carico (concentrati, ripartiti anche parzialmente e coppie) il programma disegna automaticamente i diagrammi.

PROGETTO VERIFICA SEZIONI: calcolo delle sezioni in c.a.: progetto libero o condizionato e verifica per sez. rettangolare o a T, soggetta a flessione semplice o a pressoflessione con armatura in solo zona tesa o con doppia armatura.

VERIFICA A TAGLIO: permette la verifica, in una trave, della sollecitazione di taglio ed esegue il calcolo delle staffe e dei ferri piegati a seconda della normativa vigente.

Tutti i programmi eseguono complete relazioni di calcolo.

Stress/86

TELAI PIANI AD ASTE INCLINATE

Il programma risolve telai ad aste inclinate e nodi comunque vincolati, può accettare un numero indefinito di condizioni di carico, si dispone automaticamente per la condizione successiva; è accettato altresì qualsiasi tipo di carico applicato sia alle aste che ai nodi. Il programma registra su supporto di massa tutti i dati relativi alla descrizione della geometria strutturale; è possibile perciò correggere i dati suddetti anche ad esecuzione ultimata. Il metodo utilizzato per la risoluzione dell'equazione è quello generale dell'analisi matriciale; le equazioni sono risolte con il metodo di Cholesky. Il programma fornisce come risultati le carat-

teristiche di sollecitazione agli estremi di ogni asta. Sequenzialmente viene eseguito il progetto verifica delle sezioni di incastro e quindi di qualsiasi altra sezione di ogni asta, tenendo conto scrupolosamente della normativa vigente.

STRUTTURE RETICOLARI PIANE

Procedura di calcolo completamente automatica di strutture reticolari piane con disegno dello schema strutturale. Il metodo di calcolo è quello dell'analisi matriciale e la numerazione dei nodi può essere qualsiasi; l'input è particolarmente semplice perché il video funziona in modo «autoriscordante»; sinteticamente il programma è diviso in 5 parti: richiesta dei dati geometrici; disegno dello schema strutturale con la numerazione dei nodi e delle aste; richiesta delle azioni esterne; calcolo degli sforzi nelle aste; calcolo delle deformazioni dei nodi; stampa della relazione di calcolo.

Cont/80

Il programma permette di tenere la contabilità con grande semplicità e costi contenuti. Pone l'elaboratore al servizio dell'utente; può essere usato senza alcuna conoscenza di elaborazione dati ed un minimo di conoscenza di ragioneria. Uso con uno o più terminali.

Riporto automatico dei saldi di conto e del piano dei conti su nuovo disco. Definizione libera, senza intervento sul programma di codice e nome conto, aliquote IVA, codici con imponibilità, esenzione IVA.

Elevato controllo su dati immessi. Visualizzazione o stampa di giornale, schede contabili, bilancio di verifica con subtotali, piano conti, registri IVA. Elenchi clienti e fornitori da allegare alla dichiarazione IVA di fine anno.

Contabilità lavori

È un package realizzato per il professionista che consente la gestione di tutta la parte contabile dei lavori e contiene i seguenti programmi:

- Calcolo della revisione dei prezzi per lavori edili;
- Computi metrici: consente di memorizzare descrizioni e prezzi delle voci che interessano e di eseguire su questo archivio correzioni e stampe al fine di ottenere computi metrici, offerte per appalti-concorso, stati di avanzamento dei lavori ecc.;
- Relazioni: relazioni di calcolo per strutture in cemento armato da allegare agli elaborati per il Genio Civile; relazioni di inizio lavori; relazioni di avanzamento dei lavori eventualmente già formattate per stampa su carta bollata ecc.

Paghe e contributi

Questa biblioteca di programmi consente la gestione del personale per tutte le categorie (edili, commercio, industria ecc.). Il programma esegue tutti i calcoli necessari, la stampa dei cedolini, mensili, la stampa dei contributi (su modello INPS), la stampa del libro paga nonché quella di tutte le scritture di fine anno necessaria alla più completa gestione del personale.

E inoltre gli altri ormai famosi...

- CAD PACK/1000 - ERPS/1000
- LABOR/1000 - CONT/1000
- EDT/1000
- SISMICO/80 - ISTUNO/80
- STRUTTURALE DUE - FONDAZIONI
- DISFER/80 - PORPLUS
- TOPOGRAFIA/80
- TERMOTECNICA - LABOR/80.

GRUPPO UNIVERS
00163 ROMA
VIA SARNINO, 64
TELEFONI:
779092
776468 - 758936

SONO INTERESSATO A: ACQUISTO HP ACQUISTO SOFTWARE LEASING
 ACQUISTO PERIFERICHE OFFERTA DEPLIANT ILLUSTRATIVI
DESIDERO RICEVERE: NOM E COGNOME PROFESSIONE
CITTA' _____ ENTE _____ ETA _____ TEL _____
SOCIETA' _____ VIA _____