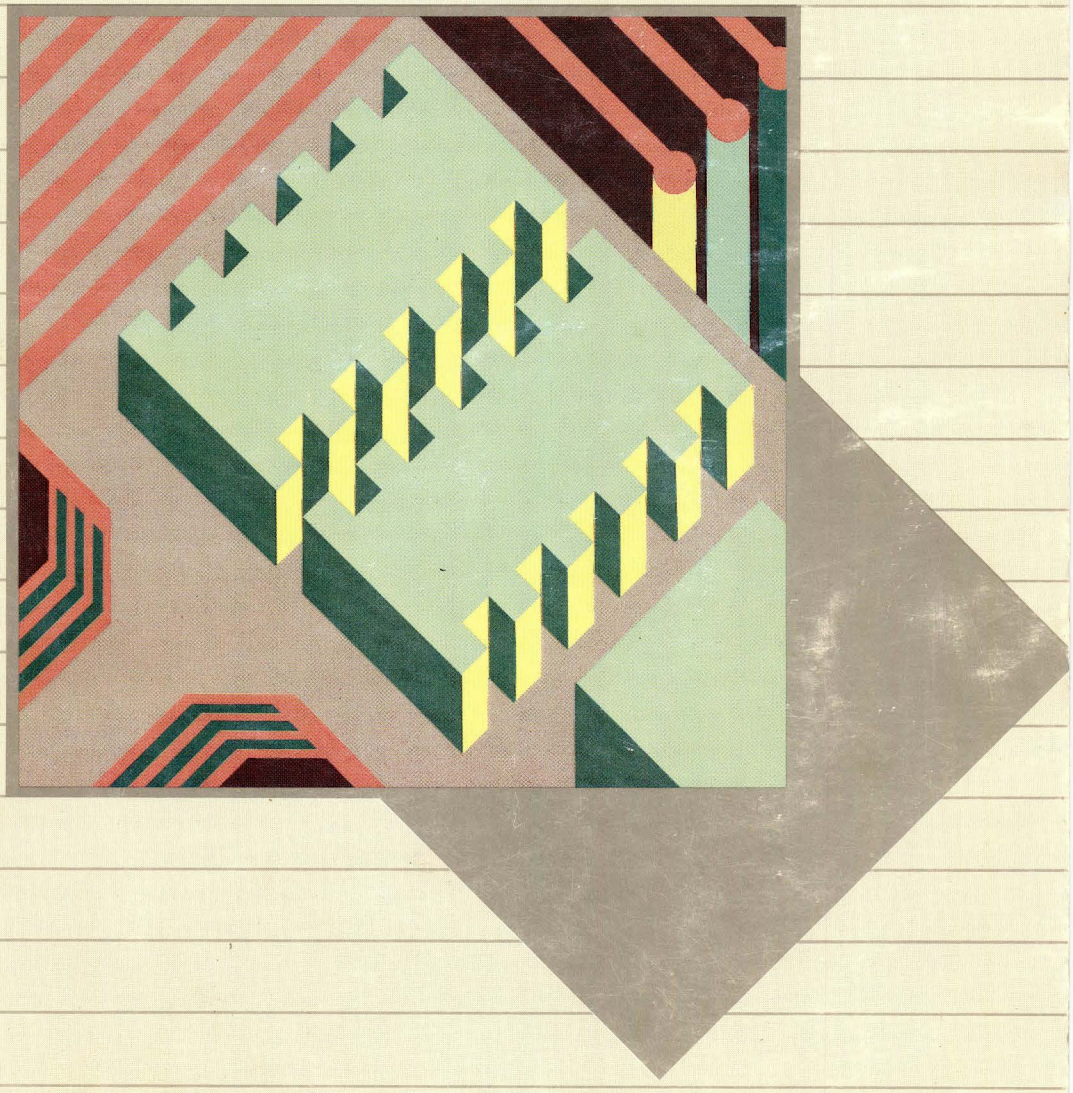
Apple //

Reference Manual Addendum: Monitor ROM Listings

For //e Only



Notice

Apple Computer, Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

Disclaimer of All Warranties and Liabilities

Apple Computer, Inc. makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. Apple Computer, Inc. software is sold or licensed "as is." The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer (and not Apple Computer, Inc., its distributor, or its retailer) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will Apple Computer, Inc. be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software, even if Apple Computer, Inc. has been advised of the possiblity of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010

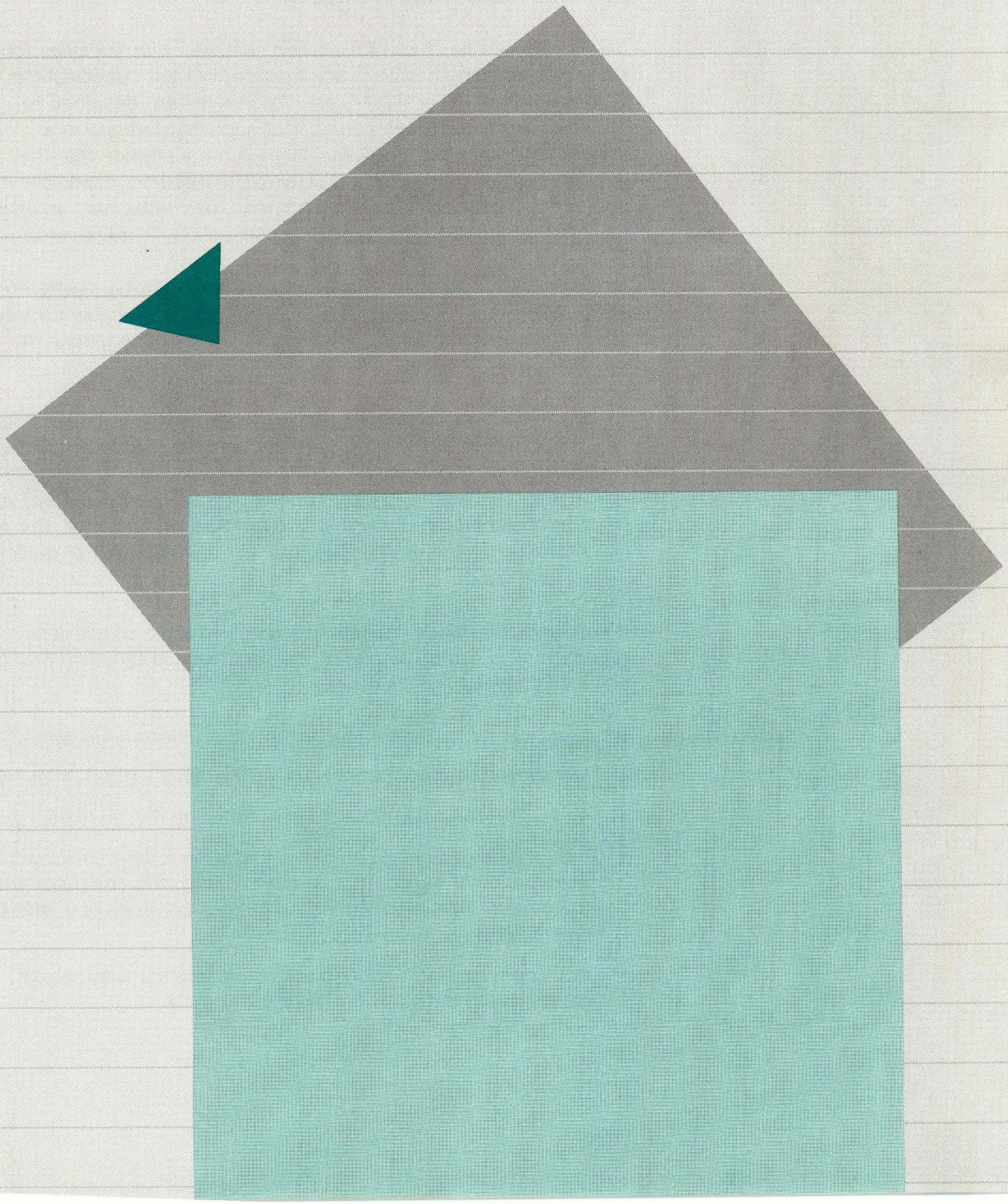
Simultaneously published in the U.S.A and Canada.

Warning

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Apple *II*

Reference Manual Addendum: Monitor ROM Listings



Radio and Television Interference

The equipment described in this manual generates and uses radio-frequency energy. If it is not installed and used properly, that is, in strict accordance with our instructions, it may cause interference with radio and television reception.

This equipment has been tested and complies with the limits for a Class B computing device in accordance with the specifications in Subpart J, Part 15, of FCC rules. These rules are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation, especially if you use a "rabbit ear" television antenna. (A "rabbit ear" antenna is the telescoping-rod type usually contained on TV receivers.)

You can determine whether your computer is causing interference by turning it off. If the interference stops, it was probably caused by the computer or its peripheral devices. To further isolate the problem:

- Disconnect the peripheral devices and their input/output cables one at a time. If the interference stops, it is caused by either the peripheral device or its I/O cable. These devices usually require shielded I/O cables. For Apple peripheral devices, you can obtain the proper shielded cable from your dealer. For non-Apple peripheral devices, contact the manufacturer or dealer for assistance.

If your computer does cause interference to radio or television reception, you can try to correct the interference by using one or more of the following measures:

- Turn the TV or radio antenna until the interference stops.
- Move the computer to one side or the other of the TV or radio.
- Move the computer farther away from the TV or radio.
- Plug the computer into an outlet that is on a different circuit than the TV or radio. (That is, make certain the computer and the radio or television set are on circuits controlled by different circuit breakers or fuses.)
- Consider installing a rooftop television antenna with coaxial cable lead-in between the antenna and TV.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find helpful the following booklet, prepared by the Federal Communications Commission:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, stock number 004-000-00345-4.

Table of Contents

3 Monitor ROM Listings

Monitor Firmware Listing

```
0000:                    2 ********************************
0000:                    3 *
0000:                    4 * APPLE II
0000:                    5 * MONITOR II
0000:                    6 *
0000:                    7 * COPYRIGHT 1978 BY
0000:                    8 * APPLE COMPUTER, INC.
0000:                    9 *
0000:                   10 * ALL RIGHTS RESERVED
0000:                   11 *
0000:                   12 * STEVE WOZNIAK
0000:                   13 *
0000:                   14 ********************************
0000:                   15 *
0000:                   16 * MODIFIED NOV 1978
0000:                   17 * BY JOHN A
0000:                   18 *
0000:                   19 * MODIFIED SEP 1981
0000:                   20 * BY RICK AURICCHIO
0000:                   21 *    & BRYAN STEARNS
0000:                   22 *    FOR APPLE2E 80COLS
0000:                   23 *
0000:                   24 * CHANGES MARKED BY 'RRA0981'
0000:                   25 *
0000:          0001     26 APPLE2E      EQU   1             ;COND ASSM/RRA0981
0000:                   27 *
0000:                   28 ********************************
----- NEXT OBJECT FILE NAME IS BJS.SRC1.OBJ0
F800:          F800     29              ORG   $F800
F800:          0000     30              OBJ   $2000
F800:                   31 ********************************
F800:          0000     32 LOC0         EQU   $00
F800:          0001     33 LOC1         EQU   $01
F800:          0020     34 WNDLFT       EQU   $20
F800:          0021     35 WNDWDTH      EQU   $21
F800:          0022     36 WNDTOP       EQU   $22
F800:          0023     37 WNDBTM       EQU   $23
F800:          0024     38 CH           EQU   $24
F800:          0025     39 CV           EQU   $25
F800:          0026     40 GBASL        EQU   $26
F800:          0027     41 GBASH        EQU   $27
F800:          0028     42 BASL         EQU   $28
F800:          0029     43 BASH         EQU   $29
F800:          002A     44 BAS2L        EQU   $2A
F800:          002B     45 BAS2H        EQU   $2B
F800:          002C     46 H2           EQU   $2C
F800:          002C     47 LMNEM        EQU   $2C
F800:          002D     48 V2           EQU   $2D
F800:          002D     49 RMNEM        EQU   $2D
F800:          002E     50 MASK         EQU   $2E
F800:          002E     51 CHKSUM       EQU   $2E
F800:          002E     52 FORMAT       EQU   $2E
F800:          002F     53 LASTIN       EQU   $2F
F800:          002F     54 LENGTH       EQU   $2F
F800:          002F     55 SIGN         EQU   $2F
F800:          0030     56 COLOR        EQU   $30
F800:          0031     57 MODE         EQU   $31
F800:          0032     58 INVFLG       EQU   $32
F800:          0033     59 PROMPT       EQU   $33
F800:          0034     60 YSAV         EQU   $34
F800:          0035     61 YSAV1        EQU   $35
F800:          0036     62 CSWL         EQU   $36
F800:          0037     63 CSWH         EQU   $37
F800:          0038     64 KSWL         EQU   $38
F800:          0039     65 KSWH         EQU   $39
F800:          003A     66 PCL          EQU   $3A
F800:          003B     67 PCH          EQU   $3B
F800:          003C     68 A1L          EQU   $3C
F800:          003D     69 A1H          EQU   $3D
F800:          003E     70 A2L          EQU   $3E
F800:          003F     71 A2H          EQU   $3F
F800:          0040     72 A3L          EQU   $40
F800:          0041     73 A3H          EQU   $41
F800:          0042     74 A4L          EQU   $42
F800:          0043     75 A4H          EQU   $43
F800:          0044     76 A5L          EQU   $44
```

```
F800:          0045   77 A5H          EQU   $45
F800:          0045   78 ACC          EQU   $45          ; NOTE OVERLAP WITH A5H!
F800:          0046   79 XREG         EQU   $46
F800:          0047   80 YREG         EQU   $47
F800:          0048   81 STATUS       EQU   $48
F800:          0049   82 SPNT         EQU   $49
F800:          004E   83 RNDL         EQU   $4E
F800:          004F   84 RNDH         EQU   $4F
F800:          0095   85 PICK         EQU   $95
F800:          0200   86 IN           EQU   $0200
F800:          03F0   87 BRKV         EQU   $3F0         ; NEW VECTOR FOR BRK
F800:          03F2   88 SOFTEV       EQU   $3F2         ; VECTOR FOR WARM START
F800:          03F4   89 PWREDUP      EQU   $3F4         ; THIS MUST = EOR #$A5 OF SOFTEV+1
F800:          03F5   90 AMPERV       EQU   $3F5         ; APPLESOFT & EXIT VECTOR
F800:          03F8   91 USRADR       EQU   $03F8
F800:          03FB   92 NMI          EQU   $03FB
F800:          03FE   93 IRQLOC       EQU   $3FE
F800:          0400   94 LINE1        EQU   $400
F800:          07F8   95 MSLOT        EQU   $07F8
F800:          C000   96 IOADR        EQU   $C000
F800:          C000   97 KBD          EQU   $C000
F800:          C010   98 KBDSTRB      EQU   $C010
F800:          C020   99 TAPEOUT      EQU   $C020
F800:          C030  100 SPKR         EQU   $C030
F800:          C050  101 TXTCLR       EQU   $C050
F800:          C051  102 TXTSET       EQU   $C051
F800:          C052  103 MIXCLR       EQU   $C052
F800:          C053  104 MIXSET       EQU   $C053
F800:          C054  105 LOWSCR       EQU   $C054
F800:          C055  106 HISCR        EQU   $C055
F800:          C056  107 LORES        EQU   $C056
F800:          C057  108 HIRES        EQU   $C057
F800:          C058  109 SETANO       EQU   $C058
F800:          C059  110 CLRANO       EQU   $C059
F800:          C05A  111 SETAN1       EQU   $C05A
F800:          C05B  112 CLRAN1       EQU   $C05B
F800:          C05C  113 SETAN2       EQU   $C05C
F800:          C05D  114 CLRAN2       EQU   $C05D
F800:          C05E  115 SETAN3       EQU   $C05E
F800:          C05F  116 CLRAN3       EQU   $C05F
F800:          C060  117 TAPEIN       EQU   $C060
F800:          C064  118 PADDL0       EQU   $C064
F800:          C070  119 PTRIG        EQU   $C070
F800:          CFFF  120 CLRROM       EQU   $CFFF
F800:          E000  121 BASIC        EQU   $E000
F800:          E003  122 BASIC2       EQU   $E003
F800: 4A             123 PLOT         LSR   A            ;Y-COORD/2
F801: 08             124              PHP                ;SAVE LSB IN CARRY
F802: 20 47 F8       125              JSR   GBASCALC     ;CALC BASE ADR IN GBASL,H
F805: 28             126              PLP                ;RESTORE LSB FROM CARRY
F806: A9 0F          127              LDA   #$0F         ;MASK $0F IF EVEN
F808: 90 02   F80C   128              BCC   RTMASK
F80A: 69 E0          129              ADC   #$E0         ;MASK $F0 IF ODD
F80C: 85 2E          130 RTMASK       STA   MASK
F80E: B1 26          131 PLOT1        LDA   (GBASL),Y    ;DATA
F810: 45 30          132              EOR   COLOR        ; XOR COLOR
F812: 25 2E          133              AND   MASK         ;  AND MASK
F814: 51 26          134              EOR   (GBASL),Y    ;   XOR DATA
F816: 91 26          135              STA   (GBASL),Y    ;    TO DATA
F818: 60             136              RTS
F819: 20 00 F8       137 HLINE        JSR   PLOT         ;PLOT SQUARE
F81C: C4 2C          138 HLINE1       CPY   H2           ;DONE?
F81E: B0 11   F831   139              BCS   RTS1         ; YES, RETURN
F820: C8             140              INY                ; NO, INCR INDEX (X-COORD)
F821: 20 0E F8       141              JSR   PLOT1        ;PLOT NEXT SQUARE
F824: 90 F6   F81C   142              BCC   HLINE1       ;ALWAYS TAKEN
F826: 69 01          143 VLINEZ       ADC   #$01         ;NEXT Y-COORD
F828: 48             144 VLINE        PHA                ; SAVE ON STACK
F829: 20 00 F8       145              JSR   PLOT         ; PLOT SQUARE
F82C: 68             146              PLA
F82D: C5 2D          147              CMP   V2           ;DONE?
F82F: 90 F5   F826   148              BCC   VLINEZ       ; NO, LOOP.
F831: 60             149 RTS1         RTS
F832: A0 2F          150 CLRSCR       LDY   #$2F         ;MAX Y, FULL SCRN CLR
F834: D0 02   F838   151              BNE   CLRSC2       ;ALWAYS TAKEN
F836: A0 27          152 CLRTOP       LDY   #$27         ;MAX Y, TOP SCRN CLR
F838: 84 2D          153 CLRSC2       STY   V2           ;STORE AS BOTTOM COORD
F83A                 154                                FOR VLINE CALLS
F83A: A0 27          155              LDY   #$27         ;RIGHTMOST X-COORD (COLUMN)
F83C: A9 00          156 CLRSC3       LDA   #$00         ;TOP COORD FOR VLINE CALLS
F83E: 85 30          157              STA   COLOR        ;CLEAR COLOR (BLACK)
F840: 20 28 F8       158              JSR   VLINE        ;DRAW VLINE
F843: 88             159              DEY                ;NEXT LEFTMOST X-COORD
F844: 10 F6   F83C   160              BPL   CLRSC3       ;LOOP UNTIL DONE.
F846: 60             161              RTS
F847: 48             162 GBASCALC     PHA                ;FOR INPUT  OODEFGH
F848: 4A             163              LSR   A
F849: 29 03          164              AND   #$03
F84B: 09 04          165              ORA   #$04         ;GENERATE GBASH=000001FG
F84D: 85 27          166              STA   GBASH
F84F: 68             167              PLA                ;AND GBASL=HDEDE000
F850: 29 18          168              AND   #$18
```

```
F852:90 02   F856  169              BCC   GBCALC
F854:69 7F         170              ADC   #$7F
F856:85 26         171 GBCALC       STA   GBASL
F858:0A            172              ASL   A
F859:0A            173              ASL   A
F85A:05 26         174              ORA   GBASL
F85C:85 26         175              STA   GBASL
F85E:60            176              RTS
F85F:A5 30         177 NXTCOL       LDA   COLOR          ;INCREMENT COLOR BY 3
F861:18            178              CLC
F862:69 03         179              ADC   #$03
F864:29 0F         180 SETCOL       AND   #$0F           ;SETS COLOR=17*A MOD 16
F866:85 30         181              STA   COLOR
F868:0A            182              ASL   A              ;BOTH HALF BYTES OF COLOR EQUAL
F869:0A            183              ASL   A
F86A:0A            184              ASL   A
F86B:0A            185              ASL   A
F86C:05 30         186              ORA   COLOR
F86E:85 30         187              STA   COLOR
F870:60            188              RTS
F871:4A            189 SCRN         LSR   A              ;READ SCREEN Y-COORD/2
F872:08            190              PHP                  ;SAVE LSB (CARRY)
F873:20 47 F8      191              JSR   GBASCALC       ;CALC BASE ADDRESS
F876:B1 26         192              LDA   (GBASL),Y      ;GET BYTE
F878:28            193              PLP                  ;RESTORE LSB FROM CARRY
F879:90 04   F87F  194 SCRN2        BCC   RTMSKZ         ;IF EVEN, USE LO H
F87B:4A            195              LSR   A
F87C:4A            196              LSR   A
F87D:4A            197              LSR   A              ;SHIFT HIGH HALF BYTE DOWN
F87E:4A            198              LSR   A
F87F:29 0F         199 RTMSKZ       AND   #$0F           ;MASK 4-BITS
F881:60            200              RTS
F882:A6 3A         201 INSDS1       LDX   PCL            ;PRINT PCL,H
F884:A4 3B         202              LDY   PCH
F886:20 96 FD      203              JSR   PRYX2
F889:20 48 F9      204              JSR   PRBLNK         ;FOLLOWED BY A BLANK
F88C:A1 3A         205 INSDS2       LDA   (PCL,X)        ;GET OPCODE
F88E:A8            206              TAY
F88F:4A            207              LSR   A              ;EVEN/ODD TEST
F890:90 09   F89B  208              BCC   IEVEN
F892:6A            209              ROR   A              ;BIT 1 TEST
F893:B0 10   F8A5  210              BCS   ERR            ;XXXXXX11 INVALID OP
F895:C9 A2         211              CMP   #$A2
F897:F0 0C   F8A5  212              BEQ   ERR            ;OPCODE $89 INVALID
F899:29 87         213              AND   #$87           ;MASK BITS
F89B:4A            214 IEVEN        LSR   A              ;LSB INTO CARRY FOR L/R TEST
F89C:AA            215              TAX
F89D:BD 62 F9      216              LDA   FMT1,X         ;GET FORMAT INDEX BYTE
F8A0:20 79 F8      217              JSR   SCRN2          ;R/L H-BYTE ON CARRY
F8A3:D0 04   F8A9  218              BNE   GETFMT
F8A5:A0 80         219 ERR          LDY   #$80           ;SUBSTITUTE $80 FOR INVALID OPS
F8A7:A9 00         220              LDA   #$00           ;SET PRINT FORMAT INDEX TO 0
F8A9:AA            221 GETFMT       TAX
F8AA:BD A6 F9      222              LDA   FMT2,X         ;INDEX INTO PRINT FORMAT TABLE
F8AD:85 2E         223              STA   FORMAT         ;SAVE FOR ADR FIELD FORMATTING
F8AF:29 03         224              AND   #$03           ;MASK FOR 2-BIT LENGTH
F8B1:              225 ; (0=1 BYTE, 1=2 BYTE, 2=3 BYTE)
F8B1:85 2F         226              STA   LENGTH
F8B3:98            227              TYA                  ;OPCODE
F8B4:29 8F         228              AND   #$8F           ;MASK FOR 1XXX1010 TEST
F8B6:AA            229              TAX                  ; SAVE IT
F8B7:98            230              TYA                  ;OPCODE TO A AGAIN
F8B8:A0 03         231              LDY   #$03
F8BA:E0 8A         232              CPX   #$8A
F8BC:F0 0B   F8C9  233              BEQ   MNNDX3
F8BE:4A            234 MNNDX1       LSR   A
F8BF:90 08   F8C9  235              BCC   MNNDX3         ;FORM INDEX INTO MNEMONIC TABLE
F8C1:4A            236              LSR   A
F8C2:4A            237 MNNDX2       LSR   A              ;  1) 1XXX1010 => 001O1XXX
F8C3:09 20         238              ORA   #$20           ;  2) XXXYYY01 => 00111XXX
F8C5:88            239              DEY                  ;  3) XXXYYY10 => 00110XXX
F8C6:D0 FA   F8C2  240              BNE   MNNDX2         ;  4) XXXYY100 => 00100XXX
F8C8:C8            241              INY                  ;  5) XXXXX000 => 000XXXXX
F8C9:88            242 MNNDX3       DEY
F8CA:D0 F2   F8BE  243              BNE   MNNDX1
F8CC:60            244              RTS
F8CD:FF FF FF      245              DFB   $FF,$FF,$FF
F8D0:20 82 F8      246 INSTDSP      JSR   INSDS1         ;GEN FMT, LEN BYTES
F8D3:48            247              PHA                  ;SAVE MNEMONIC TABLE INDEX
F8D4:B1 3A         248 PRNTOP       LDA   (PCL),Y
F8D6:20 DA FD      249              JSR   PRBYTE
F8D9:A2 01         250              LDX   #$01           ;PRINT 2 BLANKS
F8DB:20 4A F9      251 PRNTBL       JSR   PRBL2
F8DE:C4 2F         252              CPY   LENGTH         ;PRINT INST (1-3 BYTES)
F8E0:C8            253              INY                  ;IN A 12 CHR FIELD
F8E1:90 F1   F8D4  254              BCC   PRNTOP
F8E3:A2 03         255              LDX   #$03           ;CHAR COUNT FOR MNEMONIC INDEX
F8E5:C0 04         256              CPY   #$04
F8E7:90 F2   F8DB  257              BCC   PRNTBL
F8E9:68            258              PLA                  ;RECOVER MNEMONIC INDEX
F8EA:A8            259              TAY
F8EB:B9 C0 F9      260              LDA   MNEML,Y
```

```
F8EE:85 2C        261             STA   LMNEM       ;FETCH 3-CHAR MNEMONIC
F8F0:B9 00 FA     262             LDA   MNEMR,Y     ;   (PACKED INTO 2-BYTES)
F8F3:85 2D        263             STA   RMNEM
F8F5:A9 00        264 PRMN1       LDA   #$00
F8F7:A0 05        265             LDY   #$05
F8F9:06 2D        266 PRMN2       ASL   RMNEM       ;SHIFT 5 BITS OF CHARACTER INTO A
F8FB:26 2C        267             ROL   LMNEM
F8FD:2A           268             ROL   A           ;   (CLEARS CARRY)
F8FE:88           269             DEY
F8FF:D0 F8   F8F9 270             BNE   PRMN2
F901:69 BF        271             ADC   #$BF        ;ADD "?" OFFSET
F903:20 ED FD     272             JSR   COUT        ;OUTPUT A CHAR OF MNEM
F906:CA           273             DEX
F907:D0 EC   F8F5 274             BNE   PRMN1
F909:20 48 F9     275             JSR   PRBLNK      ;OUTPUT 3 BLANKS
F90C:A4 2F        276             LDY   LENGTH
F90E:A2 06        277             LDX   #$06        ;CNT FOR 6 FORMAT BITS
F910:E0 03        278 PRADR1      CPX   #$03
F912:F0 1C   F930 279             BEQ   PRADR5      ;IF X=3 THEN ADDR.
F914:06 2E        280 PRADR2      ASL   FORMAT
F916:90 0E   F926 281             BCC   PRADR3
F918:BD B3 F9     282             LDA   CHAR1-1,X
F91B:20 ED FD     283             JSR   COUT
F91E:BD B9 F9     284             LDA   CHAR2-1,X
F921:F0 03   F926 285             BEQ   PRADR3
F923:20 ED FD     286             JSR   COUT
F926:CA           287 PRADR3      DEX
F927:D0 E7   F910 288             BNE   PRADR1
F929:60           289             RTS
F92A:88           290 PRADR4      DEY
F92B:30 E7   F914 291             BMI   PRADR2
F92D:20 DA FD     292             JSR   PRBYTE
F930:A5 2E        293 PRADR5      LDA   FORMAT
F932:C9 E8        294             CMP   #$E8        ;HANDLE REL ADR MODE
F934:B1 3A        295             LDA   (PCL),Y     ;SPECIAL (PRINT TARGET,
F936:90 F2   F92A 296             BCC   PRADR4      ;   NOT OFFSET)
F938:20 56 F9     297 RELADR      JSR   PCADJ3
F93B:AA           298             TAX               ;PCL,PCH+OFFSET+1 TO A,Y
F93C:E8           299             INX
F93D:D0 01   F940 300             BNE   PRNTYX      ;+1 TO Y,X
F93F:C8           301             INY
F940:98           302 PRNTYX      TYA
F941:20 DA FD     303 PRNTAX      JSR   PRBYTE      ;OUTPUT TARGET ADR
F944:8A           304 PRNTX       TXA               ;   OF BRANCH AND RETURN
F945:4C DA FD     305             JMP   PRBYTE
F948:A2 03        306 PRBLNK      LDX   #$03        ;BLANK COUNT
F94A:A9 A0        307 PRBL2       LDA   #$A0        ;LOAD A SPACE
F94C:20 ED FD     308 PRBL3       JSR   COUT        ;OUTPUT A BLANK
F94F:CA           309             DEX
F950:D0 F8   F94A 310             BNE   PRBL2       ;LOOP UNTIL COUNT=0
F952:60           311             RTS
F953:38           312 PCADJ       SEC               ;0=1 BYTE, 1=2 BYTE,
F954:A5 2F        313 PCADJ2      LDA   LENGTH      ;   2=3 BYTE
F956:A4 3B        314 PCADJ3      LDY   PCH
F958:AA           315             TAX               ;TEST DISPLACEMENT SIGN
F959:10 01   F95C 316             BPL   PCADJ4      ;   (FOR REL BRANCH)
F95B:88           317             DEY               ;EXTEND NEG BY DECR PCH
F95C:65 3A        318 PCADJ4      ADC   PCL
F95E:90 01   F961 319             BCC   RTS2        ;PCL+LENGTH(OR DISPL)+1 TO A
F960:C8           320             INY               ;   CARRY INTO Y (PCH)
F961:60           321 RTS2        RTS
F962:             322 ; FMT1 BYTES:     XXXXXXYO INSTRS
F962:             323 ; IF Y=0          THEN LEFT HALF BYTE
F962:             324 ; IF Y=1          THEN RIGHT HALF BYTE
F962:             325 ;                   (X=INDEX)
F962:04           326 FMT1        DFB   $04
F963:20           327             DFB   $20
F964:54           328             DFB   $54
F965:30           329             DFB   $30
F966:0D           330             DFB   $0D
F967:80           331             DFB   $80
F968:04           332             DFB   $04
F969:90           333             DFB   $90
F96A:03           334             DFB   $03
F96B:22           335             DFB   $22
F96C:54           336             DFB   $54
F96D:33           337             DFB   $33
F96E:0D           338             DFB   $0D
F96F:80           339             DFB   $80
F970:04           340             DFB   $04
F971:90           341             DFB   $90
F972:04           342             DFB   $04
F973:20           343             DFB   $20
F974:54           344             DFB   $54
F975:33           345             DFB   $33
F976:0D           346             DFB   $0D
F977:80           347             DFB   $80
F978:04           348             DFB   $04
F979:90           349             DFB   $90
F97A:04           350             DFB   $04
F97B:20           351             DFB   $20
F97C:54           352             DFB   $54
```

```
F97D: 3B          353              DFB   $3B
F97E: 0D          354              DFB   $0D
F97F: 80          355              DFB   $80
F980: 04          356              DFB   $04
F981: 90          357              DFB   $90
F982: 00          358              DFB   $00
F983: 22          359              DFB   $22
F984: 44          360              DFB   $44
F985: 33          361              DFB   $33
F986: 0D          362              DFB   $0D
F987: C8          363              DFB   $C8
F988: 44          364              DFB   $44
F989: 00          365              DFB   $00
F98A: 11          366              DFB   $11
F98B: 22          367              DFB   $22
F98C: 44          368              DFB   $44
F98D: 33          369              DFB   $33
F98E: 0D          370              DFB   $0D
F98F: C8          371              DFB   $C8
F990: 44          372              DFB   $44
F991: A9          373              DFB   $A9
F992: 01          374              DFB   $01
F993: 22          375              DFB   $22
F994: 44          376              DFB   $44
F995: 33          377              DFB   $33
F996: 0D          378              DFB   $0D
F997: 80          379              DFB   $80
F998: 04          380              DFB   $04
F999: 90          381              DFB   $90
F99A: 01          382              DFB   $01
F99B: 22          383              DFB   $22
F99C: 44          384              DFB   $44
F99D: 33          385              DFB   $33
F99E: 0D          386              DFB   $0D
F99F: 80          387              DFB   $80
F9A0: 04          388              DFB   $04
F9A1: 90          389              DFB   $90
F9A2: 26          390              DFB   $26
F9A3: 31          391              DFB   $31
F9A4: 87          392              DFB   $87
F9A5: 9A          393              DFB   $9A
F9A6:             394 ; ZZXXXYO1 INSTR'S
F9A6: 00          395 FMT2         DFB   $00        ;ERR
F9A7: 21          396              DFB   $21        ;IMM
F9A8: 81          397              DFB   $81        ;Z-PAGE
F9A9: 82          398.             DFB   $82        ;ABS
F9AA: 00          399              DFB   $00        ;IMPLIED
F9AB: 00          400              DFB   $00        ;ACCUMULATOR
F9AC: 59          401              DFB   $59        ;(ZPAG,X)
F9AD: 4D          402              DFB   $4D        ;(ZPAG),Y
F9AE: 91          403              DFB   $91        ;ZPAG,X
F9AF: 92          404              DFB   $92        ;ABS,X
F9B0: 86          405              DFB   $86        ;ABS,Y
F9B1: 4A          406              DFB   $4A        ;(ABS)
F9B2: 85          407              DFB   $85        ;ZPAG,Y
F9B3: 9D          408              DFB   $9D        ;RELATIVE
F9B4: AC          409 CHAR1        DFB   $AC        ;','
F9B5: A9          410              DFB   $A9        ;')'
F9B6: AC          411              DFB   $AC        ;','
F9B7: A3          412              DFB   $A3        ;'#'
F9B8: A8          413              DFB   $A8        ;'('
F9B9: A4          414              DFB   $A4        ;'$'
F9BA: D9          415 CHAR2        DFB   $D9        ;'Y'
F9BB: 00          416              DFB   $00
F9BC: D8          417              DFB   $D8        ;'Y'
F9BD: A4          418              DFB   $A4        ;'$'
F9BE: A4          419              DFB   $A4        ;'$'
F9BF: 00          420              DFB   $00
F9C0: 1C          421 MNEML        DFB   $1C
F9C1: 8A          422              DFB   $8A
F9C2: 1C          423              DFB   $1C
F9C3: 23          424              DFB   $23
F9C4: 5D          425              DFB   $5D
F9C5: 8B          426              DFB   $8B
F9C6: 1B          427              DFB   $1B
F9C7: A1          428              DFB   $A1
F9C8: 9D          429              DFB   $9D
F9C9: 8A          430              DFB   $8A
F9CA: 1D          431              DFB   $1D
F9CB: 23          432              DFB   $23
F9CC: 9D          433              DFB   $9D
F9CD: 8B          434              DFB   $8B
F9CE: 1D          435              DFB   $1D
F9CF: A1          436              DFB   $A1
F9D0: 00          437              DFB   $00
F9D1: 29          438              DFB   $29
F9D2: 19          439              DFB   $19
F9D3: AE          440              DFB   $AE
F9D4: 69          441              DFB   $69
F9D5: A8          442              DFB   $A8
F9D6: 19          443              DFB   $19
F9D7: 23          444              DFB   $23
```

```
F9D8:24          445          DFB   $24
F9D9:53          446          DFB   $53
F9DA:1B          447          DFB   $1B
F9DB:23          448          DFB   $23
F9DC:24          449          DFB   $24
F9DD:53          450          DFB   $53
F9DE:19          451          DFB   $19          ; (A) FORMAT ABOVE
F9DF:A1          452          DFB   $A1
F9E0:00          453          DFB   $00
F9E1:1A          454          DFB   $1A
F9E2:5B          455          DFB   $5B
F9E3:5B          456          DFB   $5B
F9E4:A5          457          DFB   $A5
F9E5:69          458          DFB   $69
F9E6:24          459          DFB   $24          ; (B) FORMAT
F9E7:24          460          DFB   $24
F9E8:AE          461          DFB   $AE
F9E9:AE          462          DFB   $AE
F9EA:A8          463          DFB   $A8
F9EB:AD          464          DFB   $AD
F9EC:29          465          DFB   $29
F9ED:00          466          DFB   $00
F9EE:7C          467          DFB   $7C          ; (C) FORMAT
F9EF:00          468          DFB   $00
F9F0:15          469          DFB   $15
F9F1:9C          470          DFB   $9C
F9F2:6D          471          DFB   $6D
F9F3:9C          472          DFB   $9C
F9F4:A5          473          DFB   $A5
F9F5:69          474          DFB   $69
F9F6:29          475          DFB   $29          ; (D) FORMAT
F9F7:53          476          DFB   $53
F9F8:84          477          DFB   $84
F9F9:13          478          DFB   $13
F9FA:34          479          DFB   $34
F9FB:11          480          DFB   $11
F9FC:A5          481          DFB   $A5
F9FD:69          482          DFB   $69
F9FE:23          483          DFB   $23          ; (E) FORMAT
F9FF:A0          484          DFB   $A0
FA00:D8          485 MNEMR    DFB   $D8
FA01:62          486          DFB   $62
FA02:5A          487          DFB   $5A
FA03:48          488          DFB   $48
FA04:26          489          DFB   $26
FA05:62          490          DFB   $62
FA06:94          491          DFB   $94
FA07:88          492          DFB   $88
FA08:54          493          DFB   $54
FA09:44          494          DFB   $44
FA0A:C8          495          DFB   $C8
FA0B:54          496          DFB   $54
FA0C:68          497          DFB   $68
FA0D:44          498          DFB   $44
FA0E:E8          499          DFB   $E8
FA0F:94          500          DFB   $94
FA10:00          501          DFB   $00
FA11:B4          502          DFB   $B4
FA12:08          503          DFB   $08
FA13:84          504          DFB   $84
FA14:74          505          DFB   $74
FA15:B4          506          DFB   $B4
FA16:28          507          DFB   $28
FA17:6E          508          DFB   $6E
FA18:74          509          DFB   $74
FA19:F4          510          DFB   $F4
FA1A:CC          511          DFB   $CC
FA1B:4A          512          DFB   $4A
FA1C:72          513          DFB   $72
FA1D:F2          514          DFB   $F2
FA1E:A4          515          DFB   $A4          ; (A) FORMAT
FA1F:8A          516          DFB   $8A
FA20:00          517          DFB   $00
FA21:AA          518          DFB   $AA
FA22:A2          519          DFB   $A2
FA23:A2          520          DFB   $A2
FA24:74          521          DFB   $74
FA25:74          522          DFB   $74
FA26:74          523          DFB   $74          ; (B) FORMAT
FA27:72          524          DFB   $72
FA28:44          525          DFB   $44
FA29:68          526          DFB   $68
FA2A:B2          527          DFB   $B2
FA2B:32          528          DFB   $32
FA2C:B2          529          DFB   $B2
FA2D:00          530          DFB   $00
FA2E:22          531          DFB   $22          ; (C) FORMAT
FA2F:00          532          DFB   $00
FA30:1A          533          DFB   $1A
FA31:1A          534          DFB   $1A
FA32:26          535          DFB   $26
FA33:26          536          DFB   $26
```

```
FA34: 72                537             DFB   $72
FA35: 72                538             DFB   $72
FA36: 88                539             DFB   $88          ; (D) FORMAT
FA37: C8                540             DFB   $C8
FA38: C4                541             DFB   $C4
FA39: CA                542             DFB   $CA
FA3A: 26                543             DFB   $26
FA3B: 48                544             DFB   $48
FA3C: 44                545             DFB   $44
FA3D: 44                546             DFB   $44
FA3E: A2                547             DFB   $A2          ; (E) FORMAT
FA3F: C8                548             DFB   $C8
FA40: 85 45             549 IRQ         STA   ACC          ; *** IRQ HANDLER
FA42: 68                550             PLA
FA43: 48                551             PHA
FA44: 0A                552             ASL   A
FA45: 0A                553             ASL   A
FA46: 0A                554             ASL   A
FA47: 30 03   FA4C      555             BMI   BREAK        ;TEST FOR 'BRK'
FA49: 6C FE 03          556             JMP   (IRQLOC)     ;USER ROUTINE VECTOR IN RAM
FA4C: 28                557 BREAK       PLP
FA4D: 20 4C FF          558             JSR   SAV1         ;SAVE REG'S ON BREAK
FA50: 68                559             PLA                ; INCLUDING PC
FA51: 85 3A             560             STA   PCL
FA53: 68                561             PLA
FA54: 85 3B             562             STA   PCH
FA56: 6C F0 03          563             JMP   (BRKV)       ;BRKV WRITTEN OVER BY DISK BOOT
FA59: 20 82 F8          564 OLDBRK      JSR   INSDS1       ;PRINT USER PC
FA5C: 20 DA FA          565             JSR   RGDSP1       ;  AND REGS
FA5F: 4C 65 FF          566             JMP   MON          ;GO TO MONITOR (NO PASS GO, NO $200!)
FA62: D8                567 RESET       CLD                ;DO THIS FIRST THIS TIME
FA63: 20 84 FE          568             JSR   SETNORM
FA66: 20 2F FB          569             JSR   INIT
FA69: 20 93 FE          570             JSR   SETVID
FA6C: 20 89 FE          571             JSR   SETKBD
FA6F: AD 58 C0          572 INITAN      LDA   SETANO       ; ANO = TTL HI
FA72: AD 5A C0          573             LDA   SETAN1       ; AN1 = TTL HI
FA75:           0001    574             DO    APPLE2E      ;/RRA0981
FA75: A0 05             575             LDY   #5           ;CODE=INIT/RRA0981
FA77: 20 B4 FB          576             JSR   GOTOCX       ;DO APPLE2E INIT/RRA0981
FA7A: EA                577             NOP                ;/RRA0981
FA7B:                   578             ELSE               ;/RRA0981
 S                      579             LDA   CLRAN2       ; AN2 = TTL LO
 S                      580             LDA   CLRAN3       ; AN3 = TTL LO
FA7B:                   581             FIN                ;/RRA0981
FA7B: AD FF CF          582             LDA   CLRROM       ; TURN OFF EXTNSN ROM
FA7E: 2C 10 C0          583             BIT   KBDSTRB      ; CLEAR KEYBOARD
FA81: D8                584 NEWMON      CLD
FA82: 20 3A FF          585             JSR   BELL         ; CAUSES DELAY IF KEY BOUNCES
FA85: AD F3 03          586             LDA   SOFTEV+1     ;IS RESET HI
FA88: 49 A5             587             EOR   #$A5         ;A FUNNY COMPLEMENT OF THE
FA8A: CD F4 03          588             CMP   PWREDUP      ; PWR UP BYTE ???
FA8D: D0 17   FAA6      589             BNE   PWRUP        ; NO SO PWRUP
FA8F: AD F2 03          590             LDA   SOFTEV       ; YES SEE IF COLD START
FA92: D0 0F   FAA3      591             BNE   NOFIX        ; HAS BEEN DONE YET?
FA94: A9 E0             592             LDA   #$E0         ; DOES SOFT ENTRY VECTOR POINT AT BASIC
FA96: CD F3 03          593             CMP   SOFTEV+1
FA99: D0 08   FAA3      594             BNE   NOFIX        ; YES SO REENTER SYSTEM
FA9B: A0 03             595 FIXSEV      LDY   #3           ; NO SO POINT AT WARM START
FA9D: 8C F2 03          596             STY   SOFTEV       ; FOR NEXT RESET
FAA0: 4C 00 E0          597             JMP   BASIC        ; AND DO THE COLD START
FAA3: 6C F2 03          598 NOFIX       JMP   (SOFTEV)     ; SOFT ENTRY VECTOR
FAA6:                   599 ******************
FAA6: 20 60 FB          600 PWRUP       JSR   APPLEII
FAA9:           FAA9    601 SETPG3      EQU   *            ; SET PAGE 3 VECTORS
FAA9: A2 05             602             LDX   #5
FAAB: BD FC FA          603 SETPLP      LDA   PWRCON-1,X   ; WITH CNTRL B ADRS
FAAE: 9D EF 03          604             STA   BRKV-1,X     ; OF CURRENT BASIC
FAB1: CA                605             DEX
FAB2: D0 F7   FAAB      606             BNE   SETPLP
FAB4: A9 C8             607             LDA   #$C8         ; LOAD HI SLOT +1
FAB6: 86 00             608             STX   LOC0         ; SETPG3 MUST RETURN X=0
FAB8: 85 01             609             STA   LOC1         ; SET PTR H
FABA: A0 07             610 SLOOP       LDY   #7           ; Y IS BYTE PTR
FABC: C6 01             611             DEC   LOC1
FABE: A5 01             612             LDA   LOC1
FAC0: C9 C0             613             CMP   #$C0         ; AT LAST SLOT YET?
FAC2: F0 D7   FA9B      614             BEQ   FIXSEV       ; YES AND IT CAN'T BE A DISK
FAC4: 8D F8 07          615             STA   MSLOT
FAC7: B1 00             616 NXTBYT      LDA   (LOC0),Y     ; FETCH A SLOT BYTE
FAC9: D9 01 FB          617             CMP   DISKID-1,Y   ; IS IT A DISK ??
FACC: D0 EC   FABA      618             BNE   SLOOP        ; NO, SO NEXT SLOT DOWN
FACE: 88                619             DEY
FACF: 88                620             DEY                ; YES, SO CHECK NEXT BYTE
FAD0: 10 F5   FAC7      621             BPL   NXTBYT       ; UNTIL 4 BYTES CHECKED
FAD2: 6C 00 00          622             JMP   (LOC0)       ; GO BOOT...
FAD5: EA                623             NOP
FAD6: EA                624             NOP
FAD7:                   625 * REGDSP MUST ORG $FAD7
FAD7: 20 8E FD          626 REGDSP      JSR   CROUT        ;DISPLAY USER REG CONTENTS
FADA: A9 45             627 RGDSP1      LDA   #$45         ;WITH LABELS
FADC: 85 40             628             STA   A3L
```

```
FADE: A9 00          629              LDA   #$00
FAE0: 85 41          630              STA   A3H
FAE2: A2 FB          631              LDX   #$FB
FAE4: A9 A0          632 RDSP1        LDA   #$A0
FAE6: 20 ED FD       633              JSR   COUT
FAE9: BD 1E FA       634              LDA   RTBL-251,X
FAEC: 20 ED FD       635              JSR   COUT
FAEF: A9 BD          636              LDA   #$BD
FAF1: 20 ED FD       637              JSR   COUT
FAF4:                638 * LDA ACC+5,X
FAF4: B5 4A          639              DFB   $B5,$4A
FAF6: 20 DA FD       640              JSR   PRBYTE
FAF9: E8             641              INX
FAFA: 30 EB   FAE4   642              BMI   RDSP1
FAFC: 60             643              RTS
FAFD: 59 FA          644 PWRCON       DW    OLDBRK
FAFF: 00 E0 45       645              DFB   $00,$E0,$45
FB02: 20 FF 00 FF    646 DISKID       DFB   $20,$FF,$00,$FF
FB06: 03 FF 3C       647              DFB   $03,$FF,$3C
FB09: C1 F0 F0 EC    648 TITLE        ASC   'Apple       ]['
FB11:         FB11   649 XLTBL        EQU   *
FB11: C4 C2 C1       650              DFB   $C4,$C2,$C1
FB14: FF C3          651              DFB   $FF,$C3
FB16: FF FF FF       652              DFB   $FF,$FF,$FF
FB19:                653 *  MUST ORG $FB19
FB19: C1 D8 D9       654 RTBL         DFB   $C1,$D8,$D9   ;REGISTER NAMES FOR REGDSP:
FB1C: D0 D3          655              DFB   $D0,$D3       ;'AXYPS'
FB1E: AD 70 C0       656 PREAD        LDA   PTRIG         ;TRIGGER PADDLES
FB21: A0 00          657              LDY   #$00          ;INIT COUNT
FB23: EA             658              NOP                 ;COMPENSATE FOR 1ST COUNT
FB24: EA             659              NOP
FB25: BD 64 C0       660 PREAD2       LDA   PADDLO,X      ;COUNT Y-REG EVERY 12 USEC.
FB28: 10 04   FB2E   661              BPL   RTS2D
FB2A: C8             662              INY
FB2B: D0 F8   FB25   663              BNE   PREAD2        ;EXIT AT 255 MAX
FB2D: 88             664              DEY
FB2E: 60             665 RTS2D        RTS
FB2F:                666              CHN   BJS.SRC2
FB2F:                  1 *
FB2F: A9 00            2 INIT         LDA   #$00          ;CLR STATUS FOR DEBUG SOFTWARE
FB31: 85 48            3              STA   STATUS
FB33: AD 56 C0         4              LDA   LORES
FB36: AD 54 C0         5              LDA   LOWSCR        ;INIT VIDEO MODE
FB39: AD 51 C0         6 SETTXT       LDA   TXTSET        ;SET FOR TEXT MODE
FB3C: A9 00            7              LDA   #$00          ;FULL SCREEN WINDOW
FB3E: F0 0B   FB4B     8              BEQ   SETWND
FB40: AD 50 C0         9 SETGR        LDA   TXTCLR        ;SET FOR GRAPHICS MODE
FB43: AD 53 C0        10              LDA   MIXSET        ;LOWER 4 LINES AS TEXT WINDOW
FB46: 20 36 F8        11              JSR   CLRTOP
FB49: A9 14           12              LDA   #$14
FB4B: 85 22           13 SETWND       STA   WNDTOP        ;SET FOR 40 COL WINDOW
FB4D: A9 00           14              LDA   #$00          ;TOP IN A-REG,
FB4F: 85 20           15              STA   WNDLFT        ; BOTTOM AT LINE $24
FB51:         0001    16              DO    APPLE2E       ;/RRA0981
FB51: A0 08           17              LDY   #8            ;CODE=SETWND /RRA0981
FB53: D0 5F   FBB4    18              BNE   GOTOCX        ;DO 40/80 /RRA0981
FB55:                 19              ELSE                ;/RRA0981
 S                    20              LDA   #$28
 S                    21              STA   WNDWDTH
FB55:                 22              FIN                 ;/RRA0981
FB55: A9 18           23              LDA   #$18
FB57: 85 23           24              STA   WNDBTM
FB59: A9 17           25              LDA   #$17          ;VTAB TO ROW 23
FB5B: 85 25           26 TABV         STA   CV            ;VTABS TO ROW IN A-REG
FB5D: 4C 22 FC        27              JMP   VTAB
FB60: 20 58 FC        28 APPLEII      JSR   HOME          ;CLEAR THE SCRN
FB63: A0 08           29              LDY   #8
FB65: B9 08 FB        30 STITLE       LDA   TITLE-1,Y     ;GET A CHAR
FB68: 99 0E 04        31              STA   LINE1+14,Y    ;PUT IT AT TOP CENTER OF SCREEN
FB6B: 88              32              DEY
FB6C: D0 F7   FB65    33              BNE   STITLE
FB6E: 60              34              RTS
FB6F: AD F3 03        35 SETPWRC      LDA   SOFTEV+1      ;ROUTINE TO CALCULATE THE 'FUNNY
FB72: 49 A5           36              EOR   #$A5          ;COMPLEMENT' FOR THE RESET VECTOR
FB74: 8D F4 03        37              STA   PWREDUP
FB77: 60              38              RTS
FB78:         FB78    39 VIDWAIT      EQU   *             ;CHECK FOR A PAUSE (CONTROL-S).
FB78: C9 8D           40              CMP   #$8D          ;ONLY WHEN I HAVE A CR
FB7A: D0 18   FB94    41              BNE   NOWAIT        ;NOT SO, DO REGULAR
FB7C: AC 00 C0        42              LDY   KBD           ;IS KEY PRESSED?
FB7F: 10 13   FB94    43              BPL   NOWAIT        ;NO.
FB81: C0 93           44              CPY   #$93          ;YES -- IS IT CTRL-S?
FB83: D0 0F   FB94    45              BNE   NOWAIT        ;NOPE - IGNORE
FB85: 2C 10 C0        46              BIT   KBDSTRB       ;CLEAR STROBE
FB88: AC 00 C0        47 KBDWAIT      LDY   KBD           ;WAIT TILL NEXT KEY TO RESUME
FB8B: 10 FB   FB88    48              BPL   KBDWAIT       ;WAIT FOR KEYPRESS
FB8D: C0 83           49              CPY   #$83          ;IS IT CONTROL-C?
FB8F: F0 03   FB94    50              BEQ   NOWAIT        ;YES, SO LEAVE IT
FB91: 2C 10 C0        51              BIT   KBDSTRB       ;CLR STROBE
FB94: 4C FD FB        52 NOWAIT       JMP   VIDOUT        ;DO AS BEFORE
FB97: 38              53 ESCOLD       SEC                 ;INSURE CARRY SET
```

```
FB98:4C 2C FC       54              JMP   ESC1
FB9B:A8             55 ESCNOW       TAY                   ;USE CHAR AS INDEX
FB9C:B9 48 FA       56              LDA   XLTBL-$C9,Y     ;TRANSLATE IJKM TO CBAD
FB9F:20 97 FB       57              JSR   ESCOLD          ;DO THE CURSOR MOTION
FBA2:         0001  58              DO    APPLE2E         ;FOR FULL ESCAPES/RRA0981
FBA2:20 21 FD       59              JSR   RDESC           ;GET IJKM, ijkm, ARROWS/RRA0981
FBA5:               60              ELSE                  ;/RRA0981
 S                  61              JSR   RDKEY           ;AND GET NEXT
FBA5:               62              FIN                   ;/RRA0981
FBA5:C9 CE          63 ESCNEW       CMP   #$CE            ;IS THIS AN 'N'?
FBA7:B0 EE    FB97  64              BCS   ESCOLD          ;'N' OR GREATER - DO IT!
FBA9:C9 C9          65              CMP   #$C9            ;LESS THAN 'I'?
FBAB:90 EA    FB97  66              BCC   ESCOLD          ;YES, SO DO OLD WAY
FBAD:C9 CC          67              CMP   #$CC            ;IS IT AN 'L'?
FBAF:F0 E6    FB97  68              BEQ   ESCOLD          ;DO NORMAL
FBB1:D0 E8    FB9B  69              BNE   ESCNOW          ;GO DO IT
FBB3:         0001  70              DO    APPLE2E         ;/RRA0981
FBB3:         C006  71 SETSLOTCXROM EQU   $C006           ;/RRA0981
FBB3:         C007  72 SETINTCXROM  EQU   $C007           ;/RRA0981
FBB3:         C015  73 RDCXROM      EQU   $C015           ;/RRA0981
FBB3:               74 *                                  /RRA0981
FBB3:06             75 VERSION      DFB   $06             ;FOR IDCHECK/RRA0981
FBB4:         FBB4  76 GOTOCX       EQU   *               ;/RRA0981
FBB4:08             77              PHP                   ;SAVE USER IRQ STATE/RRA0981
FBB5:78             78              SEI                   ;INHIBIT DURING BANKSWITCH/RRA0981
FBB6:2C 15 C0       79              BIT   RDCXROM         ;GET CURRENT STATE/RRA0981
FBB9:08             80              PHP                   ;SAVE ROMBANK STATE/RRA0981
FBBA:8D 07 C0       81              STA   SETINTCXROM     ;SET ROMS ON/RRA0981
FBBD:4C 00 C1       82              JMP   $C100           ;=>OFF TO CXSPACE/RRA0981
FBC0:               83 *                                  /RRA0981
FBC0:EA             84              NOP                   ;/RRA0981
FBC1:               85              ELSE                  ;/RRA0981
 S                  86              NOP                   ;"I GOT PLENTY OF NOTHING!"
 S                  87              NOP
 S                  88              NOP
 S                  89              NOP
 S                  90              NOP
 S                  91              NOP
 S                  92              NOP
 S                  93              NOP
 S                  94              NOP
 S                  95              NOP
 S                  96              NOP
 S                  97              NOP
 S                  98              NOP
 S                  99              NOP
FBC1:               100             FIN                   ;/RRA0981
FBC1:               101 *    MUST ORG $FBC1
FBC1:48             102 BASCALC     PHA                   ;CALCBASE ADDR IN BASL,H
FBC2:4A             103             LSR   A               ; FOR GIVEN LINE NO.
FBC3:29 03          104             AND   #$03            ; 0<=LINE NO.<=$17
FBC5:09 04          105             ORA   #$04            ;ARG=000ABCDE, GENERATE
FBC7:85 29          106             STA   BASH            ; BASH=000001CD
FBC9:68             107             PLA                   ; AND
FBCA:29 18          108             AND   #$18            ; BASL=EABAB000
FBCC:90 02    FBD0  109             BCC   BASCLC2
FBCE:69 7F          110             ADC   #$7F
FBD0:85 28          111 BASCLC2     STA   BASL
FBD2:0A             112             ASL   A
FBD3:0A             113             ASL   A
FBD4:05 28          114             ORA   BASL
FBD6:85 28          115             STA   BASL
FBD8:60             116             RTS
FBD9:C9 87          117 BELL1       CMP   #$87            ;BELL CHAR? (CONTROL-G)
FBDB:D0 12    FBEF  118             BNE   RTS2B           ; NO, RETURN.
FBDD:A9 40          119             LDA   #$40            ; YES...
FBDF:20 A8 FC       120             JSR   WAIT            ;DELAY .01 SECONDS
FBE2:A0 C0          121             LDY   #$C0
FBE4:A9 0C          122 BELL2       LDA   #$0C            ;TOGGLE SPEAKER AT 1 KHZ
FBE6:20 A8 FC       123             JSR   WAIT            ; FOR .1 SEC.
FBE9:AD 30 C0       124             LDA   SPKR
FBEC:88             125             DEY
FBED:D0 F5    FBE4  126             BNE   BELL2
FBEF:60             127 RTS2B       RTS
FBF0:A4 24          128 STORADV     LDY   CH              ;CURSOR H INDEX TO Y-REG
FBF2:91 28          129             STA   (BASL),Y        ;STORE CHAR IN LINE
FBF4:E6 24          130 ADVANCE     INC   CH              ;INCREMENT CURSOR H INDEX
FBF6:A5 24          131             LDA   CH              ; (MOVE RIGHT)
FBF8:C5 21          132             CMP   WNDWDTH         ;BEYOND WINDOW WIDTH?
FBFA:B0 66    FC62  133             BCS   CR              ; YES, CR TO NEXT LINE.
FBFC:60             134 RTS3        RTS                   ; NO, RETURN.
FBFD:C9 A0          135 VIDOUT      CMP   #$A0            ;CONTROL CHAR?
FBFF:B0 EF    FBF0  136             BCS   STORADV         ; NO, OUTPUT IT.
FC01:AB             137             TAY                   ;INVERSE VIDEO?
FC02:10 EC    FBF0  138             BPL   STORADV         ; YES, OUTPUT IT.
FC04:C9 8D          139             CMP   #$8D            ;CR?
FC06:F0 5A    FC62  140             BEQ   CR              ; YES.
FC08:C9 8A          141             CMP   #$8A            ;LINE FEED?
FC0A:F0 5A    FC66  142             BEQ   LF              ; IF SO, DO IT.
FC0C:C9 88          143             CMP   #$88            ;BACK SPACE? (CONTROL-H)
FC0E:D0 C9    FBD9  144             BNE   BELL1           ; NO, CHECK FOR BELL.
FC10:C6 24          145 BS          DEC   CH              ;DECREMENT CURSOR H INDEX
```

```
FC12: 10 E8    FBFC   146              BPL   RTS3          ;IF POSITIVE, OK; ELSE MOVE UP.
FC14: A5 21           147              LDA   WNDWDTH       ;SET CH TO WINDOW WIDTH - 1.
FC16: 85 24           148              STA   CH
FC18: C6 24           149              DEC   CH            ;(RIGHTMOST SCREEN POS)
FC1A: A5 22           150 UP           LDA   WNDTOP        ;CURSOR V INDEX
FC1C: C5 25           151              CMP   CV
FC1E: B0 0B    FC2B   152              BCS   RTS4          ;IF TOP LINE THEN RETURN
FC20: C6 25           153              DEC   CV            ;DECR CURSOR V INDEX
FC22: A5 25           154 VTAB         LDA   CV            ;GET CURSOR V INDEX
FC24: 20 C1 FB        155 VTABZ        JSR   BASCALC       ;GENERATE BASE ADDRESS
FC27: 65 20           156              ADC   WNDLFT        ;ADD WINDOW LEFT OFFSET
FC29: 85 28           157              STA   BASL          ; TO BASL
FC2B: 60              158 RTS4         RTS
FC2C: 49 C0           159 ESC1         EOR   #$C0          ;ESC '@'?
FC2E: F0 28    FC58   160              BEQ   HOME          ; IF SO DO HOME AND CLEAR
FC30: 69 FD           161              ADC   #$FD          ;ESC-A OR B CHECK
FC32: 90 C0    FBF4   162              BCC   ADVANCE       ; A, ADVANCE
FC34: F0 DA    FC10   163              BEQ   BS            ; B, BACKSPACE
FC36: 69 FD           164              ADC   #$FD          ;ESC-C OR D CHECK
FC38: 90 2C    FC66   165              BCC   LF            ; C, DOWN
FC3A: F0 DE    FC1A   166              BEQ   UP            ; D, GO UP
FC3C: 69 FD           167              ADC   #$FD          ;ESC-E OR F CKECK
FC3E: 90 5C    FC9C   168              BCC   CLREOL        ; E, CLEAR TO END OF LINE
FC40: D0 E9    FC2B   169              BNE   RTS4          ; ELSE NOT F,RETURN
FC42:          0001   170              DO    APPLE2E       ;/RRA0981
FC42:          FC42   171 CLREOP       EQU   *             ;/RRA0981
FC42: A0 00           172              LDY   #0            ;CODE=CLREOP/RRA0981
FC44: F0 2C    FC72   173              BEQ   XGOTOCX       ;DO 40/80 /RRA0981
FC46: A8 C3 A9 A0     174              ASC   '(C)  1981-82, APPLE'
FC58:                 175              ELSE                ;/RRA0981
 S                    176 CLREOP       LDY   CH            ;ESC F IS CLR TO END OF PAGE
 S                    177              LDA   CV
 S                    178 CLEOP1       PHA                 ;SAVE CURRENT LINE NO.  ON STACK
 S                    179              JSR   VTABZ         ;CALC BASE ADDRESS
 S                    180              JSR   CLEOLZ        ;CLEAR TO EOL.  (SETS CARRY)
 S                    181              LDY   #$00          ;CLEAR FROM H INDEX=0 FOR REST
 S                    182              PLA                 ;INCREMENT CURRENT LINE NO.
 S                    183              ADC   #$00          ;(CARRY IS STILL SET)
 S                    184              CMP   WNDBTM        ;DONE TO BOTTOM OF WINDOW?
 S                    185              BCC   CLEOP1        ; NO, KEEP CLEARING LINES.
 S                    186              BCS   VTAB          ; YES, VTAB TO CURRENT LINE.
FC58:                 187              FIN                 ;/RRA0981
FC58:          0001   188              DO    APPLE2E       ;/RRA0981
FC58:          FC58   189 HOME         EQU   *             ;/RRA0981
FC58: A0 01           190              LDY   #1            ;CODE=HOME/RRA0981
FC5A: D0 16    FC72   191              BNE   XGOTOCX       ;DO 40/80 /RRA0981
FC5C: D2 C9 C3 CB     192              ASC   'RICK  A'     ;OUR HERO...
FC62:                 193              ELSE                ;/RRA0981
 S                    194 HOME         LDA   WNDTOP        ;INIT CURSOR V
 S                    195              STA   CV            ;  AND H INDICES
 S                    196              LDY   #$00
 S                    197              STY   CH            ;THEN CLEAR TO END OF PAGE.
 S                    198              BEQ   CLEOP1        ;(ALWAYS TAKEN)
FC62:                 199              FIN                 ;/RRA0981
FC62: A9 00           200 CR           LDA   #$00          ;CURSOR TO LEFT OF INDEX
FC64: 85 24           201              STA   CH            ;(RET CURSOR H=0)
FC66: E6 25           202 LF           INC   CV            ;INCR CURSOR V.  (DOWN 1 LINE)
FC68: A5 25           203              LDA   CV
FC6A: C5 23           204              CMP   WNDBTM        ;OFF SCREEN?
FC6C: 90 B6    FC24   205              BCC   VTABZ         ;  NO, SET BASE ADDR
FC6E: C6 25           206              DEC   CV            ;DECR CURSOR V.  (BACK TO BOTTOM)
FC70:          0001   207              DO    APPLE2E       ;/RRA0981
FC70:          FC70   208 SCROLL       EQU   *             ;/RRA0981
FC70: A0 02           209              LDY   #2            ;CODE=SCROLL/RRA0981
FC72: 4C B4 FB        210 XGOTOCX      JMP   GOTOCX        ;DO 40/80 /RRA0981
FC75:                 211 *
FC75:                 212 * IRQ SNIFFER FOR VIDEO CODE:
FC75:                 213 *
FC75:          C018   214 RDB0STORE    EQU   $C018         ;/RRA0981
FC75:          C01C   215 RDPAGE2      EQU   $C01C         ;/RRA0981
FC75: 48              216              PHA                 ;PRESERVE AC/RRA0981
FC76: AD 18 C0        217              LDA   RDB0STORE     ;FLAG->N /RRA0981
FC79: 0A              218              ASL   A             ;FLAG->C /RRA0981
FC7A: 68              219              PLA                 ;RESTORE AC/RRA0981
FC7B: 2C 1C C0        220              BIT   RDPAGE2       ;FLAG->N /RRA0981
FC7E: 08              221              PHP                 ;/RRA0981
FC7F: 90 03    FC84   222              BCC   RDCX          ;NOT BANKSWITCHING/RRA0981
FC81: 8D 54 C0        223              STA   $C054         ;FORCE MB TXTPAGE/RRA0981
FC84: 2C 15 C0        224 RDCX         BIT   RDCXROM       ;FLAG->N /RRA0981
FC87: 8D 06 C0        225              STA   SETSLOTCXROM  ;RESTORE BANK/RRA0981
FC8A: 58              226              CLI                 ;ENABLE IRQ/RRA0981
FC8B: 78              227              SEI                 ;NOW DISABLE/RRA0981
FC8C: 10 03    FC91   228              BPL   ISSLOTS       ;=>WAS SLOTS/RRA0981
FC8E: 8D 07 C0        229              STA   SETINTCXROM   ;BANK-IN CX/RRA0981
FC91:          FC91   230 ISSLOTS      EQU   *             ;/RRA0981
FC91: 28              231              PLP                 ;WHAT VID BANK/PAGE?/RRA0981
FC92: 90 05    FC99   232              BCC   ISPAGE1       ;=>NOT BANKED/RRA0981
FC94: 10 03    FC99   233              BPL   ISPAGE1       ;IT'S PAGE1/RRA0981
FC96: 2C 55 C0        234              BIT   $C055         ;FORCE PAGE2/RRA0981
FC99:          FC99   235 ISPAGE1      EQU   *             ;/RRA0981
FC99: 60              236              RTS                 ;CONTINUE VIDEO/RRA0981
```

```
FC9A: EA              237              NOP                ;/RRA0981
FC9B: EA              238              NOP                ;/RRA0981
FC9C:                 239              ELSE               ;/RRA0981
 S                    240 SCROLL       LDA   WNDTOP       ;START AT TOP OF SCROLL WINDOW
 S                    241              PHA
 S                    242              JSR   VTABZ        ;GENERATE BASE ADDRESS
 S                    243 SCRL1        LDA   BASL         ;COPY BASL,H
 S                    244              STA   BAS2L        ; TO BAS2L,H
 S                    245              LDA   BASH
 S                    246              STA   BAS2H
 S                    247              LDY   WNDWDTH      ;INIT Y TO RIGHTMOST INDEX
 S                    248              DEY                ; OF SCROLLING WINDOW
 S                    249              PLA
 S                    250              ADC   #$01         ;INCR LINE NUMBER
 S                    251              CMP   WNDBTM       ;DONE?
 S                    252              BCS   SCRL3        ; YES, FINISH
 S                    253              PHA
 S                    254              JSR   VTABZ        ;FORM BASL,H (BASE ADDR)
 S                    255 SCRL2        LDA   (BASL),Y     ;MOVE A CHAR UP ONE LINE
 S                    256              STA   (BAS2L),Y
 S                    257              DEY                ;NEXT CHAR OF LINE
 S                    258              BPL   SCRL2
 S                    259              BMI   SCRL1        ;NEXT LINE
 S                    260 SCRL3        LDY   #$00         ;CLEAR BOTTOM LINE
 S                    261              JSR   CLEOLZ       ;GET BASE ADDR FOR BOTTOM LINE
 S                    262              BCS   VTAB         ;CARRY IS SET
FC9C:                 263              FIN                ;/RRA0981
FC9C:          0001   264              DO    APPLE2E      ;/RRA0981
FC9C:          FC9C   265 CLREOL        EQU   *            ;/RRA0981
FC9C: 1B              266              CLC                ;SAY 'EOL'/RRA0981
FC9D: B0              267              DFB   $B0          ;'BCS' OPCODE /RRA0981
FC9E:          FC9E   268 CLREOLZ       EQU   *            ;/RRA0981
FC9E: 38              269              SEC                ;SAY 'EOLZ'/RRA0981
FC9F: 84 1F           270              STY   $1F          ;VIDEO'S YSAV1/RRA0981
FCA1: A0 03           271              LDY   #3           ;CODE=EOL/RRA0981
FCA3: 90 CD   FC72    272              BCC   XGOTOCX      ;->IT'S EOL/RRA0981
FCA5: C8              273              INY                ;CODE=EOLZ/RRA0981
FCA6: D0 CA   FC72    274              BNE   XGOTOCX      ;->ALWAYS/RRA0981
FCA8:                 275              ELSE               ;/RRA0981
 S                    276 CLREOL       LDY   CH           ;CURSOR H INDEX
 S                    277 CLEOLZ       LDA   #$A0         ;STORE BLANKS FROM 'HERE'
 S                    278 CLEOL2       STA   (BASL),Y     ;  TO END OF LINES (WNDWDTH)
 S                    279              INY
 S                    280              CPY   WNDWDTH
 S                    281              BCC   CLEOL2
 S                    282              RTS
FCA8:                 283              FIN                ;/RRA0981
FCA8: 38              284 WAIT         SEC
FCA9: 48              285 WAIT2        PHA
FCAA: E9 01           286 WAIT3        SBC   #$01
FCAC: D0 FC   FCAA    287              BNE   WAIT3        ;1.0204 USEC
FCAE: 68              288              PLA                ;(13+2712*A+512*A*A)
FCAF: E9 01           289              SBC   #$01
FCB1: D0 F6   FCA9    290              BNE   WAIT2
FCB3: 60              291              RTS
FCB4: E6 42           292 NXTA4        INC   A4L          ;INCR 2-BYTE A4
FCB6: D0 02   FCBA    293              BNE   NXTA1        ; AND A1
FCB8: E6 43           294              INC   A4H
FCBA: A5 3C           295 NXTA1        LDA   A1L          ;INCR 2-BYTE A1.
FCBC: C5 3E           296              CMP   A2L          ; AND COMPARE TO A2
FCBE: A5 3D           297              LDA   A1H          ; (CARRY SET IF >=)
FCC0: E5 3F           298              SBC   A2H
FCC2: E6 3C           299              INC   A1L
FCC4: D0 02   FCC8    300              BNE   RTS4B
FCC6: E6 3D           301              INC   A1H
FCC8: 60              302 RTS4B        RTS
FCC9: A0 4B           303 HEADR        LDY   #$4B         ;WRITE A*256 'LONG 1'
FCCB: 20 DB FC        304              JSR   ZERDLY       ; HALF CYCLES
FCCE: D0 F9   FCC9    305              BNE   HEADR        ; (650 USEC EACH)
FCD0: 69 FE           306              ADC   #$FE
FCD2: B0 F5   FCC9    307              BCS   HEADR        ;THEN A 'SHORT 0'
FCD4: A0 21           308              LDY   #$21         ; (400 USEC)
FCD6: 20 DB FC        309 WRBIT        JSR   ZERDLY       ;WRITE TWO HALF CYCLES
FCD9: C8              310              INY                ; OF 250 USEC ('0')
FCDA: C8              311              INY                ; OR 500 USEC ('1')
FCDB: 88              312 ZERDLY       DEY
FCDC: D0 FD   FCDB    313              BNE   ZERDLY
FCDE: 90 05   FCE5    314              BCC   WRTAPE       ;Y IS COUNT FOR
FCE0: A0 32           315              LDY   #$32         ; TIMING LOOP
FCE2: 88              316 ONEDLY       DEY
FCE3: D0 FD   FCE2    317              BNE   ONEDLY
FCE5: AC 20 C0        318 WRTAPE       LDY   TAPEOUT
FCE8: A0 2C           319              LDY   #$2C
FCEA: CA              320              DEX
FCEB: 60              321              RTS
FCEC: A2 08           322 RDBYTE       LDX   #$08         ;8 BITS TO READ
FCEE: 48              323 RDBYT2       PHA                ;READ TWO TRANSITIONS
FCEF: 20 FA FC        324              JSR   RD2BIT       ; (FIND EDGE)
FCF2: 68              325              PLA
FCF3: 2A              326              ROL   A            ;NEXT BIT
FCF4: A0 3A           327              LDY   #$3A         ;COUNT FOR SAMPLES
FCF6: CA              328              DEX
```

```
FCF7:D0 F5    FCEE  329              BNE   RDBYT2
FCF9:60             330              RTS
FCFA:20 FD FC       331 RD2BIT       JSR   RDBIT
FCFD:88             332 RDBIT        DEY                  ;DECR Y UNTIL
FCFE:AD 60 C0       333              LDA   TAPEIN         ; TAPE TRANSITION
FD01:45 2F          334              EOR   LASTIN
FD03:10 F8    FCFD  335              BPL   RDBIT
FD05:45 2F          336              EOR   LASTIN
FD07:85 2F          337              STA   LASTIN
FD09:C0 80          338              CPY   #$80           ;SET CARRY ON Y-REG.
FD0B:60             339              RTS
FD0C:A4 24          340 RDKEY        LDY   CH
FD0E:B1 28          341              LDA   (BASL),Y       ;SET SCREEN TO FLASH
FD10:48             342              PHA
FD11:29 3F          343              AND   #$3F
FD13:09 40          344              ORA   #$40
FD15:91 28          345              STA   (BASL),Y
FD17:68             346              PLA
FD18:6C 38 00       347              JMP   (KSWL)         ;GO TO USER KEY-IN
FD1B:         0001  348              DO    APPLE2E
FD1B:         FD1B  349 KEYIN        EQU   *
FD1B:A0 06          350              LDY   #6             ;RDKEY/RRA0981
FD1D:4C B4 FB       351              JMP   GOTOCX         ;/RRA0981
FD20:EA             352              NOP                  ;/RRA0981
FD21:         FD21  353 RDESC        EQU   *
FD21:20 0C FD       354              JSR   RDKEY          ;GET A KEY
FD24:A0 07          355              LDY   #7             ;CODE=FIXIT
FD26:4C B4 FB       356              JMP   GOTOCX
FD29:               357 *
FD29:               358 * RETURN FROM GOTOCX HERE:
FD29:               359 *
FD29:8D 06 C0       360              STA   SETSLOTCXROM   ;RESTORE BANK/RRA0981
FD2C:28             361              PLP                  ;RESTORE IRQ/RRA0981
FD2D:60             362              RTS                  ;RETURN TO CALLER/RRA0981
FD2E:               363              ELSE                 ;/RRA0981
 S                  364 KEYIN        INC   RNDL           ;INCR RND NUMBER
 S                  365              BNE   KEYIN2
 S                  366              INC   RNDH
 S                  367 KEYIN2       BIT   KBD            ;KEY DOWN?
 S                  368              BPL   KEYIN          ; LOOP UNTIL WE GET ONE
 S                  369              STA   (BASL),Y       ;REPLACE FLASHING SCREEN
 S                  370              LDA   KBD            ;GET KEYCODE
 S                  371              BIT   KBDSTRB        ;CLR KEY STROBE
FD2E:               372              FIN                  ;/RRA0981
FD2E:60             373              RTS
FD2F:         0001  374              DO    APPLE2E        ;/RRA0981
FD2F:20 21 FD       375 ESC          JSR   RDESC          ;/RRA0981
FD32:               376              ELSE                 ;/RRA0981
 S                  377 ESC          JSR   RDKEY          ;GET KEYCODE
FD32:               378              FIN                  ;/RRA0981
FD32:20 A5 FB       379              JSR   ESCNEW         ;HANDLE ESC FUNCTION.
FD35:20 0C FD       380 RDCHAR       JSR   RDKEY          ;GO READ KEY
FD38:C9 9B          381              CMP   #$9B           ;'ESC'?
FD3A:F0 F3    FD2F  382              BEQ   ESC            ;  YES, DON'T RETURN.
FD3C:60             383              RTS
FD3D:A5 32          384 NOTCR        LDA   INVFLG
FD3F:48             385              PHA
FD40:A9 FF          386              LDA   #$FF
FD42:         0001  387              DO    APPLE2E        ;/RRA0981
FD42:EA             388              NOP                  ;DON'T CHANGE INPUT/RRA0981
FD43:EA             389              NOP                  ; TO NORMAL/RRA0981
FD44:               390              ELSE                 ;/RRA0981
 S                  391              STA   INVFLG         ;CONVERT TYPED CHAR TO 'NORMAL'
FD44:               392              FIN                  ;/RRA0981
FD44:BD 00 02       393              LDA   IN,X
FD47:20 ED FD       394              JSR   COUT           ;ECHO TYPED CHAR
FD4A:68             395              PLA
FD4B:85 32          396              STA   INVFLG
FD4D:BD 00 02       397              LDA   IN,X
FD50:C9 88          398              CMP   #$88           ;CHECK FOR EDIT KEYS
FD52:F0 1D    FD71  399              BEQ   BCKSPC         ;  - BACKSPACE
FD54:C9 98          400              CMP   #$98
FD56:F0 0A    FD62  401              BEQ   CANCEL         ;  - CONTROL-X
FD58:E0 F8          402              CPX   #$F8
FD5A:90 03    FD5F  403              BCC   NOTCR1         ;MARGIN?
FD5C:20 3A FF       404              JSR   BELL           ; YES, SOUND BELL
FD5F:E8             405 NOTCR1       INX                  ;ADVANCE INPUT INDEX
FD60:D0 13    FD75  406              BNE   NXTCHAR
FD62:A9 DC          407 CANCEL       LDA   #$DC           ;BACKSLASH AFTER CANCELLED LINE
FD64:20 ED FD       408              JSR   COUT
FD67:20 8E FD       409 GETLNZ       JSR   CROUT          ;OUTPUT 'CR'
FD6A:A5 33          410 GETLN        LDA   PROMPT         ;OUTPUT PROMPT CHAR
FD6C:20 ED FD       411              JSR   COUT
FD6F:A2 01          412              LDX   #$01           ;INIT INPUT INDEX
FD71:8A             413 BCKSPC       TXA
FD72:F0 F3    FD67  414              BEQ   GETLNZ         ;WILL BACKSPACE TO 0
FD74:CA             415              DEX
FD75:20 35 FD       416 NXTCHAR      JSR   RDCHAR
FD78:C9 95          417              CMP   #$95           ;USE SCREEN CHAR
FD7A:D0 02    FD7E  418              BNE   CAPTST         ; FOR CONTROL-U
FD7C:B1 28          419              LDA   (BASL),Y
FD7E:C9 E0          420 CAPTST       CMP   #$E0           ;LOWER CASE?
```

```
FD80:90 02   FD84  421
FD82:        0001  422
FD82:29 FF         423
FD84:              424
 S                 425
FD84:              426
FD84:9D 00 02      427 ADDINP
FD87:C9 BD         428
FD89:D0 B2   FD3D  429
FD8B:20 9C FC      430
FD8E:A9 BD         431 CROUT
FD90:D0 5B   FDED  432
FD92:A4 3D         433 PRA1
FD94:A6 3C         434
FD96:20 8E FD      435 PRYX2
FD99:20 40 F9      436
FD9C:A0 00         437
FD9E:A9 AD         438
FDA0:4C ED FD      439
FDA3:A5 3C         440 XAM8
FDA5:09 07         441
FDA7:85 3E         442
FDA9:A5 3D         443
FDAB:85 3F         444
FDAD:A5 3C         445 MOD8CHK
FDAF:29 07         446
FDB1:D0 03   FDB6  447
FDB3:20 92 FD      448 XAM
FDB6:A9 A0         449 DATAOUT
FDB8:20 ED FD      450
FDBB:B1 3C         451
FDBD:20 DA FD      452
FDC0:20 BA FC      453
FDC3:90 E8   FDAD  454
FDC5:60            455 RTS4C
FDC6:4A            456 XAMPM
FDC7:90 EA   FDB3  457
FDC9:4A            458
FDCA:4A            459
FDCB:A5 3E         460
FDCD:90 02   FDD1  461
FDCF:49 FF         462
FDD1:65 3C         463 ADD
FDD3:48            464
FDD4:A9 BD         465
FDD6:20 ED FD      466
FDD9:68            467
FDDA:48            468 PRBYTE
FDDB:4A            469
FDDC:4A            470
FDDD:4A            471
FDDE:4A            472
FDDF:20 E5 FD      473
FDE2:68            474
FDE3:29 0F         475 PRHEX
FDE5:09 B0         476 PRHEXZ
FDE7:C9 BA         477
FDE9:90 02   FDED  478
FDEB:69 06         479
FDED:6C 36 00      480 COUT
FDF0:C9 A0         481 COUT1
FDF2:90 02   FDF6  482
FDF4:25 32         483
FDF6:84 35         484 COUTZ
FDF8:48            485
FDF9:20 78 FB      486
FDFC:68            487
FDFD:A4 35         488
FDFF:60            489
FE00:C6 34         490 BL1
FE02:F0 9F   FDA3  491
FE04:CA            492 BLANK
FE05:D0 16   FE1D  493
FE07:C9 BA         494
FE09:D0 BB   FDC6  495
FE0B:85 31         496 STOR
FE0D:A5 3E         497
FE0F:91 40         498
FE11:E6 40         499
FE13:D0 02   FE17  500
FE15:E6 41         501
FE17:60            502 RTS5
FE18:A4 34         503 SETMODE
FE1A:B9 FF 01      504
FE1D:85 31         505 SETMDZ
FE1F:60            506
FE20:A2 01         507 LT
FE22:B5 3E         508 LT2
FE24:95 42         509
FE26:95 44         510
FE28:CA            511
FE29:10 F7   FE22  512
```

```
BCC   ADDINP
DO    APPLE2E      ;/RRA0981
AND   #$FF         ;DON'T CONVERT TO UPPER CASE!/RRA0981
ELSE               ;/RRA0981
AND   #$DF         ;SHIFT TO UPPER CASE
FIN                ;/RRA0981
STA   IN,X         ;ADD TO INPUT BUFFER
CMP   #$8D
BNE   NOTCR
JSR   CLREOL       ;CLR TO EOL IF CR
LDA   #$8D
BNE   COUT         ;(ALWAYS)
LDY   A1H          ;PRINT CR,A1 IN HEX
LDX   A1L
JSR   CROUT
JSR   PRNTYX
LDY   #$00
LDA   #$AD         ;PRINT '-'
JMP   COUT
LDA   A1L
ORA   #$07         ;SET TO FINISH AT
STA   A2L          ; MOD 8=7
LDA   A1H
STA   A2H
LDA   A1L
AND   #$07
BNE   DATAOUT
JSR   PRA1
LDA   #$A0
JSR   COUT         ;OUTPUT BLANK
LDA   (A1L),Y
JSR   PRBYTE       ;OUTPUT BYTE IN HEX
JSR   NXTA1
BCC   MOD8CHK      ;NOT DONE YET. GO CHECK MOD 8
RTS                ;DONE.
LSR   A            ;DETERMINE IF MONITOR MODE IS
BCC   XAM          ; EXAMINE, ADD OR SUBTRACT
LSR   A
LSR   A
LDA   A2L
BCC   ADD
EOR   #$FF         ;FORM 2'S COMPLEMENT FOR SUBTRACT.
ADC   A1L
PHA
LDA   #$BD         ;PRINT '=', THEN RESULT
JSR   COUT
PLA
PHA                ;PRINT BYTE AS 2 HEX DIGITS
LSR   A            ; (DESTROYS A-REG)
LSR   A
LSR   A
LSR   A
JSR   PRHEXZ
PLA
AND   #$0F         ;PRINT HEX DIGIT IN A-REG
ORA   #$B0         ;LSBITS ONLY.
CMP   #$BA
BCC   COUT
ADC   #$06
JMP   (CSWL)       ;VECTOR TO USER OUTPUT ROUTINE
CMP   #$A0
BCC   COUTZ        ;DON'T OUTPUT CTRL'S INVERSE.
AND   INVFLG       ;MASK WITH INVERSE FLAG
STY   YSAV1        ;SAVE Y-REG
PHA                ;SAVE A-REG
JSR   VIDWAIT      ;OUTPUT CHR & CHECK FOR CTRL-S
PLA                ;RESTORE A-REG
LDY   YSAV1        ; AND Y-REG
RTS                ; RETURN TO SENDER...
DEC   YSAV
BEQ   XAM8
DEX                ;BLANK TO MON
BNE   SETMDZ       ;AFTER BLANK
CMP   #$BA         ;DATA STORE MODE?
BNE   XAMPM        ; NO; XAM, ADD, OR SUBTRACT.
STA   MODE         ;KEEP IN STORE MODE
LDA   A2L
STA   (A3L),Y      ;STORE AS LOW BYTE AT (A3)
INC   A3L
BNE   RTS5         ;INCR A3, RETURN.
INC   A3H
RTS
LDY   YSAV         ;SAVE CONVERTED ':', '+',
LDA   IN-1,Y       ; '-', '.' AS MODE
STA   MODE
RTS
LDX   #$01
LDA   A2L,X        ;COPY A2 (2 BYTES) TO
STA   A4L,X        ; A4 AND A5
STA   A5L,X
DEX
BPL   LT2
```

```
FE2B:60          513              RTS
FE2C:B1 3C       514 MOVE         LDA   (A1L),Y     ;MOVE (A1) THRU (A2) TO (A4)
FE2E:91 42       515              STA   (A4L),Y
FE30:20 B4 FC    516              JSR   NXTA4
FE33:90 F7  FE2C 517              BCC   MOVE
FE35:60          518              RTS
FE36:B1 3C       519 VFY          LDA   (A1L),Y     ;VERIFY (A1) THRU (A2)
FE38:D1 42       520              CMP   (A4L),Y     ; WITH (A4)
FE3A:F0 1C  FE58 521              BEQ   VFYOK
FE3C:20 92 FD    522              JSR   PRA1
FE3F:B1 3C       523              LDA   (A1L),Y
FE41:20 DA FD    524              JSR   PRBYTE
FE44:A9 A0       525              LDA   #$A0
FE46:20 ED FD    526              JSR   COUT
FE49:A9 A8       527              LDA   #$A8
FE4B:20 ED FD    528              JSR   COUT
FE4E:B1 42       529              LDA   (A4L),Y
FE50:20 DA FD    530              JSR   PRBYTE
FE53:A9 A9       531              LDA   #$A9
FE55:20 ED FD    532              JSR   COUT
FE58:20 B4 FC    533 VFYOK        JSR   NXTA4
FE5B:90 D9  FE36 534              BCC   VFY
FE5D:60          535              RTS
FE5E:20 75 FE    536 LIST         JSR   A1PC        ;MOVE A1 (2 BYTES) TO
FE61:A9 14       537              LDA   #$14        ; PC IF SPEC'D AND
FE63:48          538 LIST2        PHA               ; DISASSEMBLE 20 INSTRUCTIONS.
FE64:20 D0 F8    539              JSR   INSTDSP
FE67:20 53 F9    540              JSR   PCADJ       ;ADJUST PC AFTER EACH INSTRUCTION.
FE6A:85 3A       541              STA   PCL
FE6C:84 3B       542              STY   PCH
FE6E:68          543              PLA
FE6F:38          544              SEC
FE70:E9 01       545              SBC   #$01        ;NEXT OF 20 INSTRUCTIONS
FE72:D0 EF  FE63 546              BNE   LIST2
FE74:60          547              RTS
FE75:8A          548 A1PC         TXA               ;IF USER SPECIFIED AN ADDRESS,
FE76:F0 07  FE7F 549              BEQ   A1PCRTS     ; COPY IT FROM A1 TO PC.
FE78:B5 3C       550 A1PCLP       LDA   A1L,X       ;YEP, SO COPY IT.
FE7A:95 3A       551              STA   PCL,X
FE7C:CA          552              DEX
FE7D:10 F9  FE78 553              BPL   A1PCLP
FE7F:60          554 A1PCRTS      RTS
FE80:A0 3F       555 SETINV       LDY   #$3F        ;SET FOR INVERSE VID
FE82:D0 02  FE86 556              BNE   SETIFLG     ; VIA COUT1
FE84:A0 FF       557 SETNORM      LDY   #$FF        ;SET FOR NORMAL VID
FE86:84 32       558 SETIFLG      STY   INVFLG
FE88:60          559              RTS
FE89:A9 00       560 SETKBD       LDA   #$00        ;DO 'IN#0'
FE8B:85 3E       561 INPORT       STA   A2L         ;DO 'IN#AREG'
FE8D:A2 38       562 INPRT        LDX   #KSWL
FE8F:A0 1B       563              LDY   #KEYIN
FE91:D0 08  FE9B 564              BNE   IOPRT
FE93:A9 00       565 SETVID       LDA   #$00        ;DO 'PR#0'
FE95:85 3E       566 OUTPORT      STA   A2L         ;DO 'PR#AREG'
FE97:A2 36       567 OUTPRT       LDX   #CSWL
FE99:A0 F0       568              LDY   #COUT1
FE9B:A5 3E       569 IOPRT        LDA   A2L         ;SET INPUT/OUTPUT VECTORS
FE9D:29 0F       570              AND   #$0F
FE9F:F0 06  FEA7 571              BEQ   IOPRT1
FEA1:09 C0       572              ORA   #<IOADR
FEA3:A0 00       573              LDY   #$00
FEA5:F0 02  FEA9 574              BEQ   IOPRT2
FEA7:A9 FD       575 IOPRT1       LDA   #<COUT1
FEA9:            FEA9 576 IOPRT2  EQU   *
FEA?:94 00       577              STY   LOC0,X
FEAB:95 01       578              STA   LOC1,X
FEAD:60          579              RTS
FEAE:EA          580              NOP
FEAF:00          581 CKSUMFIX     DFB   0           ;/RRA0981
FEB0:            582 *    ;-->CORRECT CKSUM AT CREATE TIME.
FEB0:4C 00 E0    583 XBASIC       JMP   BASIC       ;TO BASIC, COLD START
FEB3:4C 03 E0    584 BASCONT      JMP   BASIC2      ;TO BASIC, WARM START
FEB6:20 75 FE    585 GO           JSR   A1PC        ;ADDR TO PC IF SPECIFIED
FEB9:20 3F FF    586              JSR   RESTORE     ;RESTORE FAKE REGISTERS
FEBC:6C 3A 00    587              JMP   (PCL)       ;AND GO!
FEBF:4C D7 FA    588 REGZ         JMP   REGDSP      ;GO DISPLAY REGISTERS
FEC2:60          589 TRACE        RTS               ;TRACE IS GONE
FEC3:EA          590              NOP
FEC4:60          591 STEPZ        RTS               ;STEP IS GONE
FEC5:C2 F2 F9 E1 592              ASC   'Bryan'
FECA:4C F8 03    593 USR          JMP   USRADR      ;JUMP TO CONTROL-Y VECTOR IN RAM
FECD:A9 40       594 WRITE        LDA   #$40        ;TAPE WRITE ROUTINE
FECF:20 C9 FC    595              JSR   HEADR       ;WRITE 10-SEC HEADER
FED2:A0 27       596              LDY   #$27
FED4:A2 00       597 WR1          LDX   #$00
FED6:41 3C       598              EOR   (A1L,X)
FED8:48          599              PHA
FED9:A1 3C       600              LDA   (A1L,X)
FEDB:20 ED FE    601              JSR   WRBYTE
FEDE:20 BA FC    602              JSR   NXTA1
FEE1:A0 1D       603              LDY   #$1D
FEE3:68          604              PLA
```

```
FEE4:90 EE   FED4  605           BCC   WR1
FEE6:A0 22         606           LDY   #$22
FEE8:20 ED FE      607           JSR   WRBYTE
FEEB:F0 4D   FF3A  608           BEQ   BELL
FEED:A2 10         609 WRBYTE    LDX   #$10
FEEF:0A            610 WRBYT2    ASL   A
FEF0:20 D6 FC      611           JSR   WRBIT
FEF3:D0 FA   FEEF  612           BNE   WRBYT2
FEF5:60            613           RTS
FEF6:20 00 FE      614 CRMON     JSR   BL1         ;HANDLE CR AS BLANK
FEF9:68            615           PLA               ; THEN POP STACK
FEFA:68            616           PLA               ; AND RETURN TO MON
FEFB:D0 6C   FF69  617           BNE   MONZ        ;(ALWAYS)
FEFD:20 FA FC      618 READ      JSR   RD2BIT      ;TAPE READ - FIND TAPEIN EDGE
FF00:A9 16         619           LDA   #$16        ;DELAY 3.5 SECONDS
FF02:20 C9 FC      620           JSR   HEADR
FF05:85 2E         621           STA   CHKSUM      ;INITIAL CHECKSUM = $FF
FF07:20 FA FC      622           JSR   RD2BIT      ;FIND AN EDGE
FF0A:A0 24         623 RD2       LDY   #$24        ;LOOK FOR SYNC BIT
FF0C:20 FD FC      624           JSR   RDBIT       ; (SHORT 0)
FF0F:B0 F9   FF0A  625           BCS   RD2         ;LOOP 'TIL FOUND
FF11:20 FD FC      626           JSR   RDBIT       ;SKIP 2ND HALF CYCLE
FF14:A0 3B         627           LDY   #$3B        ;INDEX FOR 0/1 TEST
FF16:20 EC FC      628 RD3       JSR   RDBYTE      ;READ A BYTE
FF19:81 3C         629           STA   (A1L,X)     ;PUT IT AT (A1)
FF1B:45 2E         630           EOR   CHKSUM      ;UPDATE RUNNING CHECKSUM
FF1D:85 2E         631           STA   CHKSUM
FF1F:20 BA FC      632           JSR   NXTA1       ;INCR A1, COMPARE TO A2
FF22:A0 35         633           LDY   #$35        ;COMPENSATE 0/1 INDEX
FF24:90 F0   FF16  634           BCC   RD3         ;REPEAT 'TIL DONE.
FF26:20 EC FC      635           JSR   RDBYTE      ;READ CHECKSUM BYTE
FF29:C5 2E         636           CMP   CHKSUM      ;DOES THE RECORDED CHKSM MATCH OURS?
FF2B:F0 0D   FF3A  637           BEQ   BELL        ; YEP, READ OK. BEEP AND RETURN.
FF2D:A9 C5         638 PRERR     LDA   #$C5        ;PRINT 'ERR', THEN FALL INTO
FF2F:20 ED FD      639           JSR   COUT        ; FWEEPER.
FF32:A9 D2         640           LDA   #$D2
FF34:20 ED FD      641           JSR   COUT
FF37:20 ED FD      642           JSR   COUT
FF3A:A9 87         643 BELL      LDA   #$87        ;MAKE A JOYFUL NOISE, THEN RETURN.
FF3C:4C ED FD      644           JMP   COUT
FF3F:A5 48         645 RESTORE   LDA   STATUS      ;RESTORE 6502 REGISTER CONTENTS
FF41:48            646           PHA               ; USED BY DEBUG SOFTWARE
FF42:A5 45         647           LDA   A5H
FF44:A6 46         648 RESTR1    LDX   XREG
FF46:A4 47         649           LDY   YREG
FF48:28            650           PLP
FF49:60            651           RTS
FF4A:85 45         652 SAVE      STA   A5H         ;SAVE 6502 REGISTER CONTENTS
FF4C:86 46         653 SAV1      STX   XREG        ; FOR DEBUG SOFTWARE
FF4E:84 47         654           STY   YREG
FF50:08            655           PHP
FF51:68            656           PLA
FF52:85 48         657           STA   STATUS
FF54:BA            658           TSX
FF55:86 49         659           STX   SPNT
FF57:D8            660           CLD
FF58:60            661           RTS
FF59:20 84 FE      662 OLDRST    JSR   SETNORM      ;SET SCREEN MODE
FF5C:20 2F FB      663           JSR   INIT        ; AND INIT KBD/SCREEN
FF5F:20 93 FE      664           JSR   SETVID      ; AS I/O DEVS.
FF62:20 89 FE      665           JSR   SETKBD
FF65:D8            666 MON       CLD               ;MUST SET HEX MODE!
FF66:20 3A FF      667           JSR   BELL        ;FWEEPER.
FF69:A9 AA         668 MONZ      LDA   #$AA        ;'*' PROMPT FOR MONITOR
FF6B:85 33         669           STA   PROMPT
FF6D:20 67 FD      670           JSR   GETLNZ      ;READ A LINE OF INPUT
FF70:20 C7 FF      671           JSR   ZMODE       ;CLEAR MONITOR MODE, SCAN IDX
FF73:20 A7 FF      672 NXTITM    JSR   GETNUM      ;GET ITEM, NON-HEX
FF76:84 34         673           STY   YSAV        ; CHAR IN A-REG.
FF78:A0 17         674           LDY   #$17        ; X-REG=0 IF NO HEX INPUT
FF7A:88            675 CHRSRCH   DEY
FF7B:30 E8   FF65  676           BMI   MON         ;COMMAND NOT FOUND, BEEP & TRY AGAIN.
FF7D:D9 CC FF      677           CMP   CHRTBL,Y    ;FIND COMMAND CHAR IN TABLE
FF80:D0 F8   FF7A  678           BNE   CHRSRCH     ;NOT THIS TIME
FF82:20 BE FF      679           JSR   TOSUB       ;GOT IT! CALL CORRESPONDING SUBROUTINE
FF85:A4 34         680           LDY   YSAV        ;PROCESS NEXT ENTRY ON HIS LINE
FF87:4C 73 FF      681           JMP   NXTITM
FF8A:A2 03         682 DIG       LDX   #$03
FF8C:0A            683           ASL   A
FF8D:0A            684           ASL   A           ;GOT HEX DIGIT,
FF8E:0A            685           ASL   A           ; SHIFT INTO A2
FF8F:0A            686           ASL   A
FF90:0A            687 NXTBIT    ASL   A
FF91:26 3E         688           ROL   A2L
FF93:26 3F         689           ROL   A2H
FF95:CA            690           DEX               ;LEAVE X=$FF IF DIG
FF96:10 F8   FF90  691           BPL   NXTBIT
FF98:A5 31         692 NXTBAS    LDA   MODE
FF9A:D0 06   FFA2  693           BNE   NXTBS2      ;IF MODE IS ZERO,
FF9C:B5 3F         694           LDA   A2H,X       ; THEN COPY A2 TO A1 AND A3
FF9E:95 3D         695           STA   A1H,X
FFA0:95 41         696           STA   A3H,X
```

```
FFA2:EB              697 NXTBS2      INX
FFA3:F0 F3    FF98   698             BEQ   NXTBAS
FFA5:D0 06    FFAD   699             BNE   NXTCHR
FFA7:A2 00           700 GETNUM      LDX   #$00          ;CLEAR A2
FFA9:86 3E           701             STX   A2L
FFAB:86 3F           702             STX   A2H
FFAD:B9 00 02        703 NXTCHR      LDA   IN,Y          ;GET CHAR
FFB0:C8              704             INY
FFB1:49 B0           705             EOR   #$B0
FFB3:C9 0A           706             CMP   #$0A
FFB5:90 D3    FF8A   707             BCC   DIG           ;BR IF HEX DIGIT
FFB7:69 88           708             ADC   #$88
FFB9:C9 FA           709             CMP   #$FA
FFBB:B0 CD    FF8A   710             BCS   DIG
FFBD:60              711             RTS
FFBE:A9 FE           712 TOSUB       LDA   #<GO          ;DISPATCH TO SUBROUTINE, BY
FFC0:48              713             PHA                 ; PUSHING THE HI-ORDER SUBR ADDR,
FFC1:B9 E3 FF        714             LDA   SUBTBL,Y      ; THEN THE LO-ORDER SUBR ADDR
FFC4:48              715             PHA                 ; ONTO THE STACK,
FFC5:A5 31           716             LDA   MODE          ; (CLEARING THE MODE, SAVE THE OLD
FFC7:A0 00           717 ZMODE       LDY   #$00          ; MODE IN A-REG),
FFC9:84 31           718             STY   MODE
FFCB:60              719             RTS                 ; AND 'RTS' TO THE SUBROUTINE!
FFCC:BC              720 CHRTBL      DFB   $BC           ;^C   (BASIC WARM START)
FFCD:B2              721             DFB   $B2           ;^Y   (USER VECTOR)
FFCE:BE              722             DFB   $BE           ;^E   (OPEN AND DISPLAY REGISTERS)
FFCF:B2              723             DFB   $B2           ;T    (ONCE WAS TRACE; NEVER USED.)
FFD0:EF              724             DFB   $EF           ;V    (MEMORY VERIFY)
FFD1:C4              725             DFB   $C4           ;^K   (IN#SLOT)
FFD2:B2              726             DFB   $B2           ;S    (ONCE WAS STEP; NOW NEVER USED.)
FFD3:A9              727             DFB   $A9           ;^P   (PR#SLOT)
FFD4:BB              728             DFB   $BB           ;^B   (BASIC COLD START)
FFD5:A6              729             DFB   $A6           ;'-'  (SUBTRACTION)
FFD6:A4              730             DFB   $A4           ;'+'  (ADDITION)
FFD7:06              731             DFB   $06           ;M    (MEMORY MOVE)
FFD8:95              732             DFB   $95           ;'<'  (DELIMITER FOR MOVE, VFY)
FFD9:07              733             DFB   $07           ;N    (SET NORMAL VIDEO)
FFDA:02              734             DFB   $02           ;I    (SET INVERSE VIDEO)
FFDB:05              735             DFB   $05           ;L    (DISASSEMBLE 20 INSTRS)
FFDC:F0              736             DFB   $F0           ;W    (WRITE TO TAPE)
FFDD:00              737             DFB   $00           ;G    (EXECUTE PROGRAM)
FFDE:EB              738             DFB   $EB           ;R    (READ FROM TAPE)
FFDF:93              739             DFB   $93           ;':'  (MEMORY FILL)
FFE0:A7              740             DFB   $A7           ;'.'  (ADDRESS DELIMITER)
FFE1:C6              741             DFB   $C6           ;'CR' (END OF INPUT)
FFE2:99              742             DFB   $99           ;BLANK
FFE3:B2              743 SUBTBL      DFB   $B2           ;TABLE OF LO-ORDER MONITOR ROUTINE
FFE4:C9              744             DFB   $C9           ; DISPATCH ADDRESSES
FFE5:BE              745             DFB   $BE
FFE6:C1              746             DFB   $C1
FFE7:35              747             DFB   $35
FFE8:8C              748             DFB   $8C
FFE9:C4              749             DFB   $C4
FFEA:96              750             DFB   $96
FFEB:AF              751             DFB   $AF
FFEC:17              752             DFB   $17
FFED:17              753             DFB   $17
FFEE:2B              754             DFB   $2B
FFEF:1F              755             DFB   $1F
FFF0:B3              756             DFB   $B3
FFF1:7F              757             DFB   $7F
FFF2:5D              758             DFB   $5D
FFF3:CC              759             DFB   $CC
FFF4:B5              760             DFB   $B5
FFF5:FC              761             DFB   $FC
FFF6:17              762             DFB   $17
FFF7:17              763             DFB   $17
FFF8:F5              764             DFB   $F5
FFF9:03              765             DFB   $03
FFFA:FB 03           766             DW    NMI           ;NON-MASKABLE INTERRUPT VECTOR
FFFC:62 FA           767             DW    RESET         ;RESET VECTOR
FFFE:40 FA           768             DW    IRQ           ;INTERRUPT REQUEST VECTOR
```

Monitor Symbol Table, Sorted by Symbol

```
   3D A1H          3C A1L          FE75 A1PC       FE78 A1PCLP
 FE7F A1PCRTS      3F A2H            3E A2L          41 A3H
   40 A3L          43 A4H            42 A4L          45 A5H
   44 A5L          45 ACC          FDD1 ADD        FD84 ADDINP
 FBF4 ADVANCE    ?03F5 AMPERV        01 APPLE2E    FB60 APPLEII
?  2B BAS2H     ?  2A BAS2L       FBC1 BASCALC    FBDO BASCLC2
?FEB3 BASCONT      29 BASH         E000 BASIC      E003 BASIC2
   2B BASL       FD71 BCKSPC       FBD9 BELL1      FBE4 BELL2
 FF3A BELL       FE00 BL1        ?FE04 BLANK       FA4C BREAK
 03FO BRKV       FC10 BS           FD62 CANCEL     FD7E CAPTST
 F9B4 CHAR1      F9BA CHAR2          2E CHKSUM     FF7A CHRSRCH
   24 CH         FFCC CHRTBL     ?FEAF CKSUMFIX  ?C059 CLRANO
?C05B CLRAN1   ?C05D CLRAN2     ?C05F CLRAN3      FC9C CLREOL
?FC9E CLREOLZ  ?FC42 CLREOP       CFFF CLRROM     F838 CLRSC2
 F83C CLRSC3   ?F832 CLRSCR       F836 CLRTOP       30 COLOR
 FDED COUT       FDFO COUT1        FDF6 COUTZ     FD8E CROUT
 FC62 CR       ?FEF6 CRMON       ?  37 CSWH        36 CSWL
   25 CV         FDB6 DATAOUT      FF8A DIG      FB02 DISKID
 F8A5 ERR        FD2F ESC          FC2C ESC1     F8A5 ESCNEW
 FB9B ESCNOW     F897 ESCOLD       FA9B FIXSEV   F962 FMT1
 F9A6 FMT2         2E FORMAT       F847 GBASCALC   27 GBASH
   26 GBASL      F856 GBCALC       F8A9 GETFMT   ?FD6A GETLN
 FD67 GETLNZ     FFA7 GETNUM       FBB4 GOTOCX     FEB6 GO
   2C H2         FCC9 HEADR      ?C057 HIRES     ?C055 HISCR
?F819 HLINE      F81C HLINE1       FC58 HOME     F89B IEVEN
?FA6F INITAN     FB2F INIT       ?FE8B INPORT    ?FE8D INPRT
 F882 INSDS1   ?F88C INSDS2        F8DO INSTDSP    32 INVFLG
 0200 IN         C000 IOADR        FEA7 IOPRT1   FEA9 IOPRT2
 FE9B IOPRT      03FE IRQLOC       FA40 IRQ      FC99 ISPAGE1
 FC91 ISSLOTS    C000 KBD          C010 KBDSTRB  FB8B KBDWAIT
 FD1B KEYIN   ?  39 KSWH           38 KSWL         2F LASTIN
   2F LENGTH     FC66 LF         0400 LINE1      FE63 LIST2
?FE5E LIST         2C LMNEM         00 LOCO        01 LOC1
 C056 LORES      C054 LOWSCR     ?FE20 LT        FE22 LT2
   2E MASK     ?C052 MIXCLR      C053 MIXSET     F9CO MNEML
 FA00 MNEMR      F8BE MNNDX1       F8C2 MNNDX2   F8C9 MNNDX3
 FDAD MOD8CHK      31 MODE         FF65 MON      FF69 MONZ
 FE2C MOVE       07F8 MSLOT      ?FA81 NEWMON    03FB NMI
 FAA3 NOFIX      FD5F NOTCR1       FD3D NOTCR    FB94 NOWAIT
 FCBA NXTA1      FCB4 NXTA4        FF98 NXTBAS   FF90 NXTBIT
 FFA2 NXTBS2     FAC7 NXTBYT       FD75 NXTCHAR  FFAD NXTCHR
?F85F NXTCOL     FF73 NXTITM       FA59 OLDBRK   ?FF59 OLDRST
 FCE2 ONEDLY   ?FE95 OUTPORT     ?FE97 OUTPRT    C064 PADDLO
?F954 PCADJ2     F956 PCADJ3       F953 PCADJ    F95C PCADJ4
   3B PCH          3A PCL       ?  95 PICK      F80E PLOT1
 F800 PLOT       FD92 PRA1         F910 PRADR1   F914 PRADR2
 F926 PRADR3     F92A PRADR4       F930 PRADR5   F94A PRBL2
?F94C PRBL3      F948 PRBLNK       FDDA PRBYTE   FB25 PREAD2
?FB1E PREAD   ?FF2D PRERR          FDE5 PRHEX2 ?FDE3 PRHEX
 F8F5 PRMN1      F8F9 PRMN2      ?F941 PRNTAX    F8DB PRNTBL
 FBD4 PRNTOP   ?F944 PRNTX         F940 PRNTYX     33 PROMPT
 FD96 PRYX2      C070 PTRIG        FAFD PWRCON   03F4 PWREDUP
 FAA6 PWRUP      FCFA RD2BIT       FFOA RD2      FF16 RD3
 C018 RD80STORE  FCFD RDBIT        FCEE RDBYT2   FCEC RDBYTE
 FD35 RDCHAR     C015 RDCXROM      FC84 RDCX     FD21 RDESC
 FDOC RDKEY      C01C RDPAGE2      FAE4 RDSP1  ?FEFD READ
 FAD7 REGDSP   ?FEBF REGZ        ?F938 RELADR    FA62 RESET
 FF3F RESTORE  ?FF44 RESTR1        FADA RGDSP1     2D RMNEM
?  4F RNDH     ?  4E RNDL          FB19 RTBL     FB0C RTMASK
 F87F RTMSKZ     F831 RTS1         F961 RTS2     FBEF RTS2B
 FB2E RTS2D      FBFC RTS3         FCC8 RTS4B  ?FDC5 RTS4C
 FC2B RTS4       FE17 RTS5         FF4C SAV1   ?FF4A SAVE
?F871 SCRN       F879 SCRN2      ?FC70 SCROLL    C058 SETANO
 C05A SETAN1   ?C05C SETAN2      ?C05E SETAN3  ?FB64 SETCOL
?FB40 SETGR      FE86 SETIFLG      C007 SETINTCXROM ?FE80 SETINV
 FE89 SETKBD     FE1D SETMDZ     ?FE18 SETMODE   FE84 SETNORM
?FAA9 SETPG3     FAAB SETPLP     ?FB6F SETPWRC   C006 SETSLOTCXROM
?FB39 SETTXT     FE93 SETVID       FB4B SETWND ?  2F SIGN
 FABA SLOOP      03F2 SOFTEV       C030 SPKR     49 SPNT
   48 STATUS   ?FEC4 STEPZ         FB65 STITLE ?FE0B STOR
 FBFO STORADV    FFE3 SUBTBL     ?FB5B TABV      C060 TAPEIN
 C020 TAPEOUT    FB09 TITLE        FFBE TOSUB    ?FEC2 TRACE
 C050 TXTCLR     C051 TXTSET       FC1A UP     ?FECA USR
 03F8 USRADR       2D V2         ?FBB3 VERSION   FE5B VFYOK
 FE36 VFY        FBFD VIDOUT       FB78 VIDWAIT  F828 VLINE
 F826 VLINEZ     FC22 VTAB         FC24 VTABZ    FCAB WAIT
 FCA9 WAIT2      FCAA WAIT3          23 WNDBTM     20 WNDLFT
```

```
   22 WNDTOP            21 WNDWDTH         FED4 WR1             FCD6 WRBIT
 FEEF WRBYT2          FEED WRBYTE        ?FECD WRITE           FCE5 WRTAPE
 FDB3 XAM             FDA3 XAMB           FDC6 XAMPM          ?FEB0 XBASIC
 FC72 XGOTOCX         FB11 XLTBL            46 XREQ              47 YREQ
   34 YSAV              35 YSAV1          FCDB ZERDLY          FFC7 ZMODE
** SUCCESSFUL ASSEMBLY := NO ERRORS
** ASSEMBLER CREATED ON 05-JAN-82 000004
** TOTAL LINES ASSEMBLED  1435
** FREE SPACE PAGE COUNT   67
    2  BJS.SRC2
```

Monitor Symbol Table, Sorted by Address

```
   00 LOC0          01 APPLE2E       01 LOC1          20 WNDLFT
   21 WNDWDTH       22 WNDTOP        23 WNDBTM        24 CH
   25 CV            26 GBASL         27 GBASH         28 BASL
   29 BASH        ? 2A BAS2L       ? 2B BAS2H         2C H2
   2C LMNEM         2D V2            2D RMNEM         2E MASK
   2E CHKSUM        2E FORMAT        2F LASTIN        2F LENGTH
 ? 2F SIGN          30 COLOR         31 MODE          32 INVFLG
   33 PROMPT        34 YSAV          35 YSAV1         36 CSWL
 ? 37 CSWH          38 KSWL        ? 39 KSWH          3A PCL
   3B PCH           3C A1L           3D A1H           3E A2L
   3F A2H           40 A3L           41 A3H           42 A4L
   43 A4H           44 A5L           45 A5H           45 ACC
   46 XREG          47 YREG          48 STATUS        49 SPNT
 ? 4E RNDL        ? 4F RNDH        ? 95 PICK        0200 IN
  03F0 BRKV        03F2 SOFTEV      03F4 PWREDUP    ?03F5 AMPERV
  03F8 USRADR      03FB NMI         03FE IRQLOC      0400 LINE1
  07F8 MSLOT       C000 IOADR       C000 KBD         C006 SETSLOTCXROM
  C007 SETINTCXROM C010 KBDSTRB     C015 RDCXROM     C018 RD80STORE
  C01C RDPAGE2     C020 TAPEOUT     C030 SPKR        C050 TXTCLR
  C051 TXTSET     ?C052 MIXCLR      C053 MIXSET      C054 LOWSCR
 ?C055 HISCR       C056 LORES       ?C057 HIRES      C058 SETANO
 ?C059 CLRANO      C05A SETAN1     ?C05B CLRAN1     ?C05C SETAN2
 ?C05D CLRAN2     ?C05E SETAN3     ?C05F CLRAN3      C060 TAPEIN
  C064 PADDL0      C070 PTRIG       CFFF CLRROM      E000 BASIC
  E003 BASIC2      F800 PLOT        F80C RTMASK      F80E PLOT1
 ?F819 HLINE       F81C HLINE1      F826 VLINEZ      F828 VLINE
  F831 RTS1       ?F832 CLRSCR      F836 CLRTOP      F838 CLRSC2
  F83C CLRSC3      F847 GBASCALC    F856 GBCALC     ?F85F NXTCOL
 ?F864 SETCOL     ?F871 SCRN        F879 SCRN2       F87F RTMSKZ
  F882 INSDS1     ?F88C INSDS2      F89B IEVEN       F8A5 ERR
  F8A9 GETFMT      F8BE MNNDX1      F8C2 MNNDX2      F8C9 MNNDX3
  F8D0 INSTDSP     F8D4 PRNTOP      F8DB PRNTBL      F8F5 PRMN1
  F8F9 PRMN2       F910 PRADR1      F914 PRADR2      F926 PRADR3
  F92A PRADR4      F930 PRADR5     ?F93B RELADR      F940 PRNTYX
 ?F941 PRNTAX     ?F944 PRNTX       F94B PRBLNK      F94A PRBL2
 ?F94C PRBL3       F953 PCADJ      ?F954 PCADJ2      F956 PCADJ3
  F95C PCADJ4      F961 RTS2        F962 FMT1        F9A6 FMT2
  F9B4 CHAR1       F9BA CHAR2       F9C0 MNEML       FA00 MNEMR
  FA40 IRQ         FA4C BREAK       FA59 OLDBRK      FA62 RESET
 ?FA6F INITAN     ?FA81 NEWMON      FA9B FIXSEV      FAA3 NOFIX
  FAA6 PWRUP      ?FAA9 SETPG3      FAAB SETPLP      FABA SLOOP
  FAC7 NXTBYT      FAD7 REGDSP      FADA RGDSP1      FAE4 RDSP1
  FAFD PWRCON      FB02 DISKID      FB09 TITLE       FB11 XLTBL
  FB19 RTBL       ?FB1E PREAD       FB25 PREAD2      FB2E RTS2D
  FB2F INIT       ?FB39 SETTXT     ?FB40 SETGR       FB4B SETWND
 ?FB5B TABV        FB60 APPLEII     FB65 STITLE     ?FB6F SETPWRC
  FB78 VIDWAIT     FB88 KBDWAIT     FB94 NOWAIT      FB97 ESCOLD
  FB9B ESCNOW      FBA5 ESCNEW     ?FBB3 VERSION     FBB4 GOTOCX
  FBC1 BASCALC     FBD0 BASCLC2     FBD9 BELL1       FBE4 BELL2
  FBEF RTS2B       FBF0 STORADV     FBF4 ADVANCE     FBFC RTS3
  FBFD VIDOUT      FC10 BS          FC1A UP          FC22 VTAB
  FC24 VTABZ       FC2B RTS4        FC2C ESC1       ?FC42 CLREOP
  FC58 HOME        FC62 CR          FC66 LF         ?FC70 SCROLL
  FC72 XGOTOCX     FC84 RDCX        FC91 ISSLOTS     FC99 ISPAGE1
  FC9C CLREOL     ?FC9E CLREOLZ     FCA8 WAIT        FCA9 WAIT2
  FCAA WAIT3       FCB4 NXTA4       FCBA NXTA1       FCC8 RTS4B
  FCC9 HEADR       FCD6 WRBIT       FCDB ZERDLY      FCE2 ONEDLY
  FCE5 WRTAPE      FCEC RDBYTE      FCEE RDBYT2      FCFA RD2BIT
  FCFD RDBIT       FD0C RDKEY       FD1B KEYIN       FD21 RDESC
  FD2F ESC         FD35 RDCHAR      FD3D NOTCR       FD5F NOTCR1
  FD62 CANCEL      FD67 GETLNZ     ?FD6A GETLN       FD71 BCKSPC
  FD75 NXTCHAR     FD7E CAPTST      FD84 ADDINP      FD8E CROUT
  FD92 PRA1        FD96 PRYX2       FDA3 XAMB        FDAD MOD8CHK
  FDB3 XAM         FDB6 DATAOUT    ?FDC5 RTS4C       FDC6 XAMPM
  FDD1 ADD         FDDA PRBYTE     ?FDE3 PRHEX       FDE5 PRHEXZ
  FDED COUT        FDF0 COUT1       FDF6 COUTZ       FE00 BL1
 ?FE04 BLANK      ?FE0B STOR       FE17 RTS5       ?FE18 SETMODE
  FE1D SETMDZ     ?FE20 LT         FE22 LT2         FE2C MOVE
  FE36 VFY         FE58 VFYOK      ?FE5E LIST       FE63 LIST2
  FE75 A1PC        FE78 A1PCLP      FE7F A1PCRTS    ?FE80 SETINV
  FE84 SETNORM     FE86 SETIFLG     FE89 SETKBD     ?FE8B INPORT
 ?FE8D INPRT       FE93 SETVID     ?FE95 OUTPORT    ?FE97 OUTPRT
  FE98 IOPRT       FEA7 IOPRT1      FEA9 IOPRT2     ?FEAF CKSUMFIX
 ?FEB0 XBASIC     ?FEB3 BASCONT     FEB6 GO         ?FEBF REGZ
 ?FEC2 TRACE      ?FEC4 STEPZ     ?FECA USR        ?FECD WRITE
  FED4 WR1         FEED WRBYTE      FEEF WRBYT2     ?FEF6 CRMON
```

```
?FEFD READ          FF0A RD2           FF16 RD3          ?FF2D PRERR
 FF3A BELL          FF3F RESTORE      ?FF44 RESTR1        FF4A SAVE
 FF4C SAV1         ?FF59 OLDRST        FF65 MON           FF69 MONZ
 FF73 NXTITM        FF7A CHRSRCH       FF8A DIG           FF90 NXTBIT
 FF98 NXTBAS        FFA2 NXTBS2        FFA7 GETNUM        FFAD NXTCHR
 FFBE TOSUB         FFC7 ZMODE         FFCC CHRTBL        FFE3 SUBTBL
** SUCCESSFUL ASSEMBLY := NO ERRORS
** ASSEMBLER CREATED ON 05-JAN-82 000004
** TOTAL LINES ASSEMBLED  1438
** FREE SPACE PAGE COUNT   67
    2  BJS.SRC2
```

80-Column Firmware Listing

```
0000:                 2 ****************************************
0000:                 3 *
0000:                 4 * Apple //e VIDEO FIRMWARE
0000:                 5 *
0000:                 6 * RICK AURICCHIO 08/81
0000:                 7 *
0000:                 8 * (C) 1981, APPLE COMPUTER INC.
0000:                 9 *      ALL RIGHTS RESERVED
0000:                10 *
0000:                11 ****************************************
0000:                12 *
0000:          0006  13 GOODF8       EQU   6           ;F8 ROM VERSION
0000:                14 *
0000:                15 * HARDWARE EQUATES:
0000:                16 *
0000:          C000  17 KBD          EQU   $C000       ;KEYBOARD PORT
0000:          C000  18 CLR80COL     EQU   $C000       ;DISABLE 80COL STORE
0000:          C001  19 SET80COL     EQU   $C001       ;ENABLE 80COL STORE
0000:          C002  20 RDMAINRAM    EQU   $C002       ;READ MAINBOARD RAM
0000:          C003  21 RDCARDRAM    EQU   $C003       ;READ CARD RAM
0000:          C004  22 WRMAINRAM    EQU   $C004       ;WRITE MAINBOARD RAM
0000:          C005  23 WRCARDRAM    EQU   $C005       ;WRITE CARD RAM
0000:          C007  24 SETINTCXROM  EQU   $C007       ;SET INTERNAL CX00 ROM
0000:          C008  25 SETSTDZP     EQU   $C008       ;SET STD ZP/STK
0000:          C009  26 SETALTZP     EQU   $C009       ;SET ALT ZP/STK
0000:          C00B  27 SETSLOTC3ROM EQU   $C00B
0000:          C00C  28 CLR80VID     EQU   $C00C       ;DISABLE 80COL VIDEO
0000:          C00D  29 SET80VID     EQU   $C00D       ;ENABLE 80COL VIDEO
0000:          C00E  30 CLRALTCHAR   EQU   $C00E       ;NORM LC, FLASH UC
0000:          C00F  31 SETALTCHAR   EQU   $C00F       ;NORM/INV LC, NO FLASH
0000:          C010  32 KBDSTRB      EQU   $C010       ;CLEAR STROBE
0000:          C011  33 RDLCBNK2     EQU   $C011       ;READS LC BANK2
0000:          C012  34 RDLCRAM      EQU   $C012       ;READS LC RAM ENABLE
0000:          C013  35 RDRAMRD      EQU   $C013       ;READS RAMREAD STATE
0000:          C014  36 RDRAMWRT     EQU   $C014       ;READS BANKWRT STATE
0000:          C018  37 RD80COL      EQU   $C018       ;READS SET80COL
0000:          C019  38 RDVBLBAR     EQU   $C019       ; 'VBL SIGNAL
0000:          C01A  39 RDTEXT       EQU   $C01A       ;READS TXT MODE
0000:          C01C  40 RDPAGE2      EQU   $C01C       ;PAGE1/2 STATUS
0000:          C01F  41 RD80VID      EQU   $C01F       ;READS SET80VID
0000:          C030  42 SPKR         EQU   $C030       ;TOGGLE SPEAKER
0000:          C054  43 TXTPAGE1     EQU   $C054       ;PAGE1 TEXT
0000:          C055  44 TXTPAGE2     EQU   $C055       ;PAGE2 TEXT
0000:                45 *
0000:                46 * MONITOR EQUATES:
0000:                47 *
0000:          FBB3  48 FBVERSION    EQU   $FBB3       ;F8 ROM ID
0000:          FD0C  49 RDKEY        EQU   $FD0C       ;GET A KEYSTROKE
0000:          FE89  50 SETKBD       EQU   $FE89       ;IN#0
0000:          FE93  51 SETVID       EQU   $FE93       ;PR#0
0000:          FF58  52 IORTS        EQU   $FF58       ;KNOWN RTS
0000:                54 * ZEROPAGE EQUATES:
0000:                55 *
0000:                56             DSECT
001F:                57             ORG   $1F
001F:          0001  58 YSAV1        DS    1           ;SAFE PLACE IN ALL ENVIRONS
0020:          0001  59 WNDLFT       DS    1           ;SCROLLING WINDOW LEFT
0021:          0001  60 WNDWDTH      DS    1           ;SCROLLING WINDOW WIDTH
0022:          0001  61 WNDTOP       DS    1           ;SCROLLING WINDOW TOP
0023:          0001  62 WNDBTM       DS    1           ;SCROLLING WINDOW BOTTOM
0024:          0001  63 CH           DS    1           ;CURSOR HORIZONTAL
0025:          0001  64 CV           DS    1           ;CURSOR VERTICAL
0026:          0002  65              DS    2           ;GBASL
0028:          0002  66 BASL         DS    2           ;BASE ADDRESS
002A:          0029  67 BASH         EQU   BASL+1
002A:          0002  68 BAS2L        DS    2           ;BASE ADDR FOR SCROLL
002C:          002B  69 BAS2H        EQU   BAS2L+1
0032:          0032  70              ORG   $32
0032:          0001  71 INVFLG       DS    1           ;>127=NORMAL
0033:          0003  72              DS    3           ;N/A
0036:          0002  73 CSWL         DS    2           ;COUT HOOK
0038:          0037  74 CSWH         EQU   CSWL+1
0038:          0002  75 KSWL         DS    2           ;KEYIN HOOK
003A:          0039  76 KSWH         EQU   KSWL+1
003C:          003C  77              ORG   *$3C
003C:          0002  78 A1L          DS    2           ;MONITOR TEMPS FOR MOVE
```

```
003E:          003D    79 A1H          EQU   A1L+1
003E:          0002    80 A2L          DS    2
0040:          003F    81 A2H          EQU   A2L+1
0040:          0002    82              DS    2           ;A3 NOT USED
0042:          0002    83 A4L          DS    2
0044:          0043    84 A4H          EQU   A4L+1
004E:          004E    85              ORG   $4E
004E:          0002    86 RNDL         DS    2           ;RANDOM NUMBER SEED
0050:          004F    87 RNDH         EQU   RNDL+1
0000:                  88              DEND
0000:                  90 * PERMANENT DATA IN SCREENHOLES
0000:                  91 *
0000:                  92 * NOTE: THESE RESIDE IN PAGE 1 OF
0000:                  93 *  THE 80-COLUMN SCREEN PAIR; ANY
0000:                  94 *  ROUTINE WHICH SETS PAGE2 *MUST*
0000:                  95 *  RESTORE BACK TO PAGE1 SO THAT
0000:                  96 *  WE CAN CORRECTLY ACCESS THESE
0000:                  97 *  PERMS. UNDER *NO* CIRCUMSTANCES
0000:                  98 *  IS ANY ROUTINE TO BE CALLED WHILE
0000:                  99 *  WE HAVE PAGE2 BANKED IN!
0000:                 100 *
0000:          0478   101 TEMP1        EQU   $478        ;A TEMP
0000:          047B   102 OLDCH        EQU   $478+3      ;OLD CH SET FOR USER
0000:          04FB   103 MODE         EQU   $4F8+3      ;OPERATING MODE
0000:                 104 * MODE BITS
0000:                 105 * 0.......  - ESC-R INACTIVE
0000:                 106 * 1.......  - ESC-R ACTIVE
0000:                 107 * .0......  - BASIC PRINT
0000:                 108 * .1......  - BASIC INPUT
0000:                 109 * ..0.....  - LANGUAGE=BASIC
0000:                 110 * ..1.....  - LANGUAGE=PASCAL
0000:                 111 * ...0....  - U/C RESTRICT MODE
0000:                 112 * ...1....  - LITERAL UC/LC MODE
0000:                 113 * ....0...  - GOTOXY N/A
0000:                 114 * ....1...  - GOTOXY IN PROGRESS
0000:                 115 * .....0..  - NORMAL  VIDED (PASCAL)
0000:                 116 * .....1..  - INVERSE VIDED (PASCAL)
0000:                 117 * ......0.  - PASCAL 1.1 F/W ACTIVE
0000:                 118 * ......1.  - PASCAL 1.0 INTERFACE
0000:                 119 * .......0  - CALLER SEI'D (BASIC)
0000:                 120 * .......1  - CALLER CLI'D (BASIC)
0000:                 121 *       0   - NORMAL MODE (PASCAL)
0000:                 122 *       1   - TRANSPARENT MODE (PASCAL)
0000:          0080   123 M.ESCR       EQU   $80         ;ESC-R ACTIVE
0000:          0040   124 M.BINPUT     EQU   $40         ;BASIC INPUTTING
0000:          0020   125 M.PASCAL     EQU   $20         ;PASCAL RUNNING
0000:          0010   126 M.LIT        EQU   $10         ;LITERAL UC/LC INPUT
0000:          0008   127 M.GOXY       EQU   $08         ;GOTOXY IN PROGRESS
0000:          0004   128 M.VMODE      EQU   $04         ;PASCAL VIDEO MODE
0000:          0002   129 M.PAS1.0     EQU   $02         ;PASCAL 1.0 MODE
0000:          0001   130 M.IRQ        EQU   $01         ;IRQ ENABLED (BASIC ONLY)
0000:          0001   131 M.TRANS      EQU   $01         ;TRANSPARENT MODE IF F/W PROTOCOL
0000:          057B   132 OURCH        EQU   $578+3      ;80-COL CH
0000:          05FB   133 OURCV        EQU   $5F8+3      ;CURSOR VERTICAL
0000:          067B   134 CHAR         EQU   $678+3      ;IN/OUT CHAR
0000:          06FB   135 XCOORD       EQU   $6F8+3      ;X-COORD (GOTOXY)
0000:          077B   136 OLDBASL      EQU   $778+3      ;PASCAL SAVED BASL
0000:          07FB   137 OLDBASH      EQU   $7F8+3      ;PASCAL SAVED BASH
0000:                 138 *
0000:                 139 * GENERAL SCREEN STUFF:
0000:                 140 *
0000:          07F8   141 C8SLOT       EQU   $7F8        ;IRQ C8 PROTOCOL
0000:                 142              CHR   '-'
0000:                   4              INCLUDE BFUNC
----- NEXT OBJECT FILE NAME IS VIDEO.OBJ0
C100:          C100    2              ORG   $C100
C100:          C100    3 BFUNCPG      EQU   *
C100:          FD29    4 FUNCEXIT     EQU   $FD29       ;RETURN ADDRESS
C100:          FBC1    5 F.BASCALC    EQU   $FBC1
C100:          FC22    6 F.VTAB       EQU   $FC22
C100:          FC24    7 F.VTABZ      EQU   $FC24
C100:                   8 ----------------------------------------
C100:                   9 * BASIC FUNCTION HOOK:
C100:                  10 ----------------------------------------
C100:                  11 * THIS ROUTINE IS CALLED BY THE
C100:                  12 *  PATCHED F8 ROM...
C100:                  13 *  THIS CODE WILL ALWAYS PERFORM THE
C100:                  14 *  FUNCTION HERE AND RETURN TO THE
C100:                  15 *  CALLER.
C100:                  16 *
C100:                  17 * NOTE: F8 ROM DISABLES I/O TO GET US
C100:                  18 *  RUNNING HERE.  WE RETURN TO F8 SPACE.
C100:                  19 ----------------------------------------
C100:                  20 * INPUT:  Y=FUNCTION AS FOLLOWS:
C100:                  21 *            0=CLREOP
C100:                  22 *            1=HOME
C100:                  23 *            2=SCROLL
C100:                  24 *            3=CLREOL
C100:                  25 *            4=CLEOLZ
C100:                  26 *            5=INIT @ RESET
C100:                  27 *            6=KEYIN
C100:                  28 *            7=FIX ESCAPE CHAR
```

```
C100:                    29 *           8=SETWND
C100:                    30 *
C100:                    31 *        STK HAS PHP FOR STATUS
C100:                    32 *         OF BANK & IRQ BIT
C100:                    33 * VOLATILE: AC,Y
C100:                    34 ----------------------------------------
C100:                    35 * NOTE: IF WE HAVE A CARD INSTALLED,
C100:                    36 *  THEN USE THE VIDEO ROUTINES, SINCE
C100:                    37 *  WE 'OWN' SLOT3 SCREENHOLES.
C100:                    38 * IF NOT, DUPLICATE FBROM ROUTINES
C100:                    39 *  AND AVOID SLOT3 INTERFERENCE.
C100:                    40 ----------------------------------------
C100:                    41 * VECTOR TO KEYIN/ESCFIX IMMEDIATELY
C100:                    42 *  TO AVOID AC DESTRUCTION:
C100:                    43 *
C100:          C100      44 B.FUNC       EQU   *
C100: C0 06              45              CPY   #6              ;IS IT KEYIN?
C102: D0 03    C107      46              BNE   B.FUNCNK        ;NO
C104: 4C 88 C2           47              JMP   B.KEYIN
C107:          C107      48 B.FUNCNK     EQU   *
C107: C0 07              49              CPY   #7              ;IS IT ESCAPE-FIX?
C109: D0 03    C10E      50              BNE   B.FUNCNE        ;NO
C10B: 4C 6E C2           51              JMP   B.ESCFIX        ;=>YES!
C10E:          C10E      52 B.FUNCNE     EQU   *
C10E: 98                 53              TYA                   ;SAVE Y
C10F: 48                 54              PHA
C110: 20 24 CB           55              JSR   TESTCARD        ;DO WE HAVE A CARD?
C113: D0 0A    C11F      56              BNE   B.OLDFUNC       ;=>NO
C115:                    57 *
C115:                    58 * NOTE: THIS TEST COULD TURN OUT
C115:                    59 *  WRONG ON POWER-UP, SINCE THE
C115:                    60 *  MODEBYTE IS UNDEFINED. HOWEVER,.
C115:                    61 *  SINCE THE MONITOR IS DOING A
C115:                    62 *  SIMPLE 'SETWND' CALL, WE WON'T
C115:                    63 *  GET INTO TROUBLE EVEN IF WE
C115:                    64 *  MAKE THE WRONG DECISION...
C115:                    65 *
C115: AD FB 04           66              LDA   MODE            ;IS MODE VALID?
C118: 29 28              67              AND   #M.PASCAL+M.GOXY ; FOR BASIC
C11A: D0 03    C11F      68              BNE   B.OLDFUNC       ;=>DEFINITELY NOT!
C11C: 4C A4 C1           69              JMP   B.FUNCO         ;=>YES, GO NEW WAY
C11F:                    70 *
C11F:                    71 * NO CARD.  DO THINGS THE OLD WAY.
C11F:                    72 *
C11F:          C11F      73 B.OLDFUNC    EQU   *
C11F: 68                 74              PLA
C120: A8                 75              TAY                   ;RESTORE Y
C121: A9 C1              76              LDA   #<BFUNCPG       ;TRANSFER VIA
C123: 48                 77              PHA                   ; THE RTS-TRICK
C124: B9 EA CF           78              LDA   F.TABLE,Y       ;GET LO ADDRESS
C127: 48                 79              PHA
C128: 60                 80              RTS                   ;TRANSFER TO ROUTINE
C129:                    81 ----------------------------------------
C129: A4 24              82 F.CLREOP     LDY   CH              ; ESC F IS CLR TO END OF PAGE
C12B: A5 25              83              LDA   CV
C12D: 48                 84 CLEOP1       PHA
C12E: 20 24 FC           85              JSR   F.VTABZ
C131: 20 F4 C2           86              JSR   X.CLEOLZ
C134: A0 00              87              LDY   #$00
C136: 68                 88              PLA
C137: 69 00              89              ADC   #$00
C139: C5 23              90              CMP   WNDBTM
C13B: 90 F0    C12D      91              BCC   CLEOP1
C13D: 20 22 FC           92              JSR   F.VTAB
C140: 4C EB C2           93              JMP   F.RETURN        ;DONE
C143:                    94 ----------------------------------------
C143: A5 22              95 F.HOME       LDA   WNDTOP
C145: 85 25              96              STA   CV
C147: A0 00              97              LDY   #$00
C149: 84 24              98              STY   CH
C14B: F0 E0    C12D      99              BEQ   CLEOP1          ;(ALWAYS TAKEN)
C14D:                   100 ----------------------------------------
C14D: A5 22             101 F.SCROLL     LDA   WNDTOP
C14F: 48                102              PHA
C150: 20 24 FC          103              JSR   F.VTABZ
C153: A5 28             104 SCRL1        LDA   BASL
C155: 85 2A             105              STA   BAS2L
C157: A5 29             106              LDA   BASH
C159: 85 2B             107              STA   BAS2H
C15B: A4 21             108              LDY   WNDWDTH
C15D: 88                109              DEY
C15E: 68                110              PLA
C15F: 69 01             111              ADC   #$01
C161: C5 23             112              CMP   WNDBTM
C163: B0 0D    C172     113              BCS   SCRL3
C165: 48                114              PHA
C166: 20 24 FC          115              JSR   F.VTABZ
C169: B1 28             116 SCRL2        LDA   (BASL),Y
C16B: 91 2A             117              STA   (BAS2L),Y
C16D: 88                118              DEY
C16E: 10 F9    C169     119              BPL   SCRL2
```

```
C170:30 E1    C153  120              BMI   SCRL1
C172:A0 00          121 SCRL3        LDY   #$00
C174:20 F4 C2       122              JSR   X.CLEOLZ
C177:20 22 FC       123              JSR   F.VTAB
C17A:4C EB C2       124              JMP   F.RETURN     ;=>DONE
C17D:A4 24          125 F.CLREOL     LDY   CH
C17F:A9 A0          126              LDA   #$A0
C181:91 28          127 CLEOL2       STA   (BASL),Y
C183:C8             128              INY
C184:C4 21          129              CPY   WNDWDTH
C186:90 F9    C181  130              BCC   CLEOL2
C188:B0 17    C1A1  131              BCS   F.GORET      ;DONE (ALWAYS TAKEN)
C18A:               132 ----------------------------------------
C18A:         C18A  133 F.SETWND     EQU   *
C18A:A9 28          134              LDA   #40
C18C:85 21          135              STA   WNDWDTH
C18E:A9 18          136              LDA   #24
C190:85 23          137              STA   WNDBTM
C192:A9 17          138              LDA   #23
C194:85 25          139              STA   CV
C196:20 22 FC       140              JSR   F.VTAB
C199:4C EB C2       141              JMP   F.RETURN
C19C:               142 ----------------------------------------
C19C:         C19C  143 F.CLEOLZ     EQU   *
C19C:A4 1F          144              LDY   YSAV1        ;RESTORE HORIZ POSITION
C19E:20 F4 C2       145              JSR   X.CLEOLZ     ;DO IT
C1A1:4C EB C2       146 F.GORET      JMP   F.RETURN     ;DONE
C1A4:         C1A4  148 B.FUNCO      EQU   *
C1A4:68             149              PLA                ;RESTORE Y
C1A5:A8             150              TAY
C1A6:               151 *
C1A6:               152 * SET IRQMODE:
C1A6:               153 *
C1A6:AD FB 04       154              LDA   MODE         ;ASSUME IRQ IS DISABLED
C1A9:29 FE          155              AND   #255-M.IRQ
C1AB:8D FB 04       156              STA   MODE
C1AE:68             157              PLA                ;PULL CXBANK STATUS
C1AF:8D 78 04       158              STA   TEMP1        ; OFF STACK
C1B2:68             159              PLA                ;GET USER'S PSTATUS
C1B3:48             160              PHA                ; (LEAVE ALONE ON STACK)
C1B4:4A             161              LSR   A            ;MOVE 'I' BIT TO
C1B5:4A             162              LSR   A            ; THE CARRY
C1B6:4A             163              LSR   A
C1B7:AD 78 04       164              LDA   TEMP1        ;PUT CXBANK STATUS
C1BA:48             165              PHA                ; BACK ON STACK
C1BB:B0 08    C1C5  166              BCS   NOI          ;=>HE'S INHIBITED
C1BD:AD FB 04       167              LDA   MODE
C1C0:09 01          168              ORA   #M.IRQ
C1C2:8D FB 04       169              STA   MODE
C1C5:         C1C5  170 NOI          EQU   *
C1C5:A5 25          171              LDA   CV           ;COPY USER CV
C1C7:8D FB 05       172              STA   OURCV        ; TO OURS
C1CA:4C FF C1       173              JMP   B.VECTOR     ;CONTINUE
C1CD:               174 *
C1CD:               175 * NOTE: THIS KEEPS B.XXXX ROUTINES
C1CD:               176 *  ALL IN THE C100 PAGE...
C1CD:               177 *
C1CD:               178 *
C1CD:         C1CD  179 B.SCROLL     EQU   *
C1CD:20 A4 CC       180              JSR   SCROLLUP     ;DO IT FOR CALLER
C1D0:4C EB C2       181              JMP   F.RETURN     ; AND RETURN DIRECTLY
C1D3:               182 *
C1D3:         C1D3  183 B.CLREOL     EQU   *
C1D3:20 48 CD       184              JSR   X.GS         ;CLEAR TO EOL
C1D6:4C EB C2       185              JMP   F.RETURN     ;RETURN DIRECTLY TO CALLER
C1D9:               186 ----------------------------------------
C1D9:         C1D9  187 B.CLEOLZ     EQU   *
C1D9:A4 1F          188              LDY   YSAV1        ;RESTORE HORIZ POSITION
C1DB:20 4E CD       189              JSR   X.GSEOLZ     ;DO IT TO EOL
C1DE:4C EB C2       190              JMP   F.RETURN
C1E1:               191 ----------------------------------------
C1E1:         C1E1  192 B.CLREOP     EQU   *
C1E1:20 23 CD       193              JSR   X.VT         ;CLEAR TO EOS
C1E4:4C EB C2       194              JMP   F.RETURN     ;RETURN DIRECTLY TO CALLER
C1E7:               195 ----------------------------------------
C1E7:4C 19 C2       196 B.SETWND     JMP   B.SETWNDX
C1EA:4C 34 C2       197 B.RESET      JMP   B.RESETX     ;MUST BE IN BFUNC PAGE
C1ED:               198 ----------------------------------------
C1ED:         C1ED  199 B.HOME       EQU   *
C1ED:20 42 CD       200              JSR   X.FF         ;HOME & CLEAR
C1F0:AD 7B 05       201              LDA   OURCH
C1F3:85 24          202              STA   CH           ;COPY CH/CV FOR CALLER
C1F5:8D 7B 04       203              STA   OLDCH        ;REMEMBER WHAT WE SET
C1F8:AD FB 05       204              LDA   OURCV
C1FB:85 25          205              STA   CV
C1FD:10 2F    C22E  206              BPL   GOBACK       ;(ALWAYS TAKEN)
C1FF:               207 *
C1FF:               208 * COPY USER'S CURSOR IF IT DIFFERS
C1FF:               209 *  FROM OURS (AND WE'RE RUNNING
C1FF:               210 *  IN 80-COLUMN MODE).  IF WE ARE
C1FF:               211 *  NOT IN 80-MODE, THEN ALWAYS USE
```

```
C1FF:                   212 *  THE USER'S CH VALUE SINCE OURS
C1FF:                   213 *  IS PROBABLY INVALID.
C1FF:                   214 *
C1FF:           C1FF    215 B.VECTOR    EQU   *
C1FF:20 51 CB           216             JSR   BASCALC
C202:A5 24              217             LDA   CH          ;GET USER CH VALUE
C204:2C 1F C0           218             BIT   RD80VID     ;DISPLAYING 80-COLS?
C207:10 05   C20E       219             BPL   B.GETCH     ;=>NO, USER CH IS IT
C209:CD 7B 04           220             CMP   OLDCH       ;IS IT DIFFERENT?
C20C:F0 03   C211       221             BEQ   B.FUNC1     ;=>NO, USE OURS
C20E:           C20E    222 B.GETCH     EQU   *
C20E:8D 7B 05           223             STA   OURCH       ;USE HIS CH
C211:           C211    224 B.FUNC1     EQU   *
C211:A9 C1              225             LDA   #<BFUNCPG   ;TRANSFER TO ROUTINE
C213:48                 226             PHA               ; VIA RTS-TRICK
C214:B9 F3 CF           227             LDA   B.TABLE,Y   ;GET LO ADDRESS
C217:48                 228             PHA
C218:60                 229             RTS
C219:                   230 ----------------------------------------
C219:           C219    231 B.SETWNDX   EQU   *
C219:A9 50              232             LDA   #80         ;ASSUME 80-COLS
C21B:2C 1F C0           233             BIT   RD80VID     ;WHICH MODE?
C21E:30 01   C221       234             BMI   B.SETWND2   ;=>IT'S 80
C220:4A                 235             LSR   A           ;MAKE IT 40
C221:           C221    236 B.SETWND2   EQU   *
C221:85 21              237             STA   WNDWDTH
C223:A9 1B              238             LDA   #24         ;SET BOTTOM
C225:85 23              239             STA   WNDBTM
C227:A9 17              240             LDA   #23         ;VTAB TO BOTTOM
C229:8D FB 05           241             STA   OURCV
C22C:85 25              242             STA   CV
C22E:20 51 CB           243 GOBACK      JSR   BASCALC
C231:4C EB C2           244             JMP   F.RETURN
C234:                   245 *
C234:                   246 * HANDLE RESET FOR MONITOR:
C234:                   247 *
C234:           C234    248 B.RESETX    EQU   *
C234:A9 FF              249             LDA   #$FF        ;DESTROY MODE BYTE
C236:8D FB 04           250             STA   MODE
C239:AD 5D C0           251             LDA   $C05D       ;SETUP
C23C:AD 5F C0           252             LDA   $C05F       ; ANNUNCIATORS
C23F:                   253 *
C23F:                   254 * IF THE OPEN APPLE KEY
C23F:                   255 *  (ALIAS PADDLE BUTTONS 0) IS
C23F:                   256 *  DEPRESSED, COLDSTART THE SYSTEM
C23F:                   257 *  AFTER DESTROYING MEMORY:
C23F:                   258 *
C23F:AD 62 C0           259             LDA   $C062       ;GET BUTTON 1 (SOLID)
C242:30 1D   C261       260             BMI   DIAGS       ;=>DOWN, DO DIAGS
C244:AD 61 C0           261             LDA   $C061       ;GET BUTTON 0 (OPEN)
C247:10 1B   C264       262             BPL   RESETRET    ;=>NOT JIVE OR DIAGS
C249:                   263 *
C249:                   264 * BLAST 2 BYTES OF EACH PAGE,
C249:                   265 *  INCLUDING THE RESET VECTOR:
C249:                   266 *
C249:A0 B0              267             LDY   #$B0        ;LET IT PRECESS DOWN
C24B:A9 00              268             LDA   #0
C24D:85 3C              269             STA   A1L
C24F:A9 BF              270             LDA   #$BF        ;START FROM BFXX DOWN
C251:38                 271             SEC               ;FOR SUBTRACT
C252:           C252    272 BLAST       EQU   *
C252:85 3D              273             STA   A1H
C254:91 3C              274             STA   (A1L),Y
C256:88                 275             DEY
C257:91 3C              276             STA   (A1L),Y
C259:E9 01              277             SBC   #1          ;BACK DOWN TO NEXT PAGE
C25B:C9 01              278             CMP   #1          ;STAY AWAY FROM STACK!
C25D:D0 F3   C252       279             BNE   BLAST
C25F:F0 03   C264       280             BEQ   RESETRET    ;(ALWAYS)
C261:                   281 *
C261:           C261    282 DIAGS       EQU   *
C261:4C 01 C4           283             JMP   $C401       ;RUN DIAGS
C264:                   284 *
C264:           C264    285 RESETRET    EQU   *
C264:20 24 CB           286             JSR   TESTCARD    ;CARD PLUGGED IN?
C267:F0 14   C27D       287             BEQ   GORETN      ;=>YES
C269:8D 0B C0           288             STA   SETSLOTC3ROM ;NO, DISABLE ROM
C26C:D0 0F   C27D       289             BNE   GORETN      ;(ALWAYS TAKEN)
C26E:                   290 ----------------------------------------
C26E:           C26E    291 B.ESCFIX    EQU   *
C26E:29 DF              292             AND   #$DF        ;FORCE TO UPPERCASE
C270:A0 03              293             LDY   #4-1        ;SCAN FOR A MATCH
C272:           C272    294 B.ESCFIX2   EQU   *
C272:D9 80 C2           295             CMP   ESCIN,Y     ;IS IT?
C275:D0 03   C27A       296             BNE   B.ESCFIX3   ;=>NAW
C277:B9 84 C2           297             LDA   ESCOUT,Y    ;YES, TRANSLATE IT
C27A:           C27A    298 B.ESCFIX3   EQU   *
C27A:88                 299             DEY
C27B:10 F5   C272       300             BPL   B.ESCFIX2
C27D:4C EB C2           301 GORETN      JMP   F.RETURN    ;RETURN:CHAR IN AC
C280:                   302 ----------------------------------------
```

```
C280:8B 95 8A 8B   303 ESCIN       DFB   $8B,$95,$8A,$8B
C284:CA CB CD C9   304 ESCOUT      ASC   'JKMI'          ;THE ARROWS
C288:              305 ----------------------------------------
C288:        C288  306 B.KEYIN     EQU   *
C288:BD 78 04      307              STA   TEMP1          ;SAVE ORIGINAL CHAR
C28B:68            308              PLA                  ;HOLD ONTO
C28C:A8            309              TAY                  ; CXBANK STATUS
C28D:68            310              PLA                  ;GET USER'S
C28E:48            311              PHA                  ; IRQ STATE
C28F:6A            312              ROR   A              ;MOVE IRQ BIT TO
C290:6A            313              ROR   A              ; THE
C291:6A            314              ROR   A              ;  CARRY
C292:98            315              TYA                  ;PUT CXBANK STATUS
C293:48            316              PHA                  ; BACK ON STACK
C294:8A            317              TXA                  ;SAVE
C295:48            318              PHA                  ; XREG
C296:              319 *
C296:B8            320              CLV                  ;ASSUME NOT INTERRUPTIBLE
C297:B0 03   C29C  321              BCS   B.KEYIN2       ;=>WE WERE RIGHT
C299:2C 00 CF      322              BIT   SEV            ;SAY "INTERRUPTIBLE"
C29C:        C29C  323 B.KEYIN2    EQU   *
C29C:A9 FF         324              LDA   #$FF           ;CURSOR=NORMAL DELETE
C29E:A4 24         325              LDY   CH
C2A0:91 28         326              STA   (BASL),Y
C2A2:20 C6 C2      327              JSR   KEYDLY         ;WAIT FOR A KEY
C2A5:B0 0E   C2B5  328              BCS   GOTKEY         ;=>GOT ONE
C2A7:AD 78 04      329              LDA   TEMP1          ;REPLACE ORIG CHAR
C2AA:A4 24         330              LDY   CH
C2AC:91 28         331              STA   (BASL),Y
C2AE:20 C6 C2      332              JSR   KEYDLY         ;WAIT FOR A KEY
C2B1:B0 02   C2B5  333              BCS   GOTKEY         ;=>GOT ONE
C2B3:90 E7   C29C  334              BCC   B.KEYIN2       ;(ALWAYS TAKEN)
C2B5:              335 *
C2B5:        C2B5  336 GOTKEY      EQU   *
C2B5:AD 78 04      337              LDA   TEMP1          ;RESTORE ORIGINAL
C2B8:A4 24         338              LDY   CH
C2BA:91 28         339              STA   (BASL),Y       ; CHARACTER
C2BC:68            340              PLA                  ;RESTORE
C2BD:AA            341              TAX                  ; XREG
C2BE:AD 00 C0      342              LDA   KBD            ;GET THE NEW KEYSTROKE
C2C1:8D 10 C0      343              STA   KBDSTRB        ;CANCEL THE STROBE
C2C4:30 25   C2EB  344              BMI   F.RETURN       ;(ALWAYS TAKEN)
C2C6:              345 ***
C2C6:              346 *** INPUT: VFLAG SET IF INTERRUPTIBLE
C2C6:        C2C6  347 KEYDLY      EQU   *
C2C6:A2 0C         348              LDX   #$0C           ;SHORT DELAY FOR IRQ
C2C8:70 02   C2CC  349              BVS   IK1            ;->INTERRUPTIBLE
C2CA:A2 31         350              LDX   #$31           ;LONG  DELAY FOR NO IRQ
C2CC:        C2CC  351 IK1         EQU   *
C2CC:A0 00         352              LDY   #0
C2CE:        C2CE  353 IK2         EQU   *
C2CE:50 05   C2D5  354              BVC   IK2A           ;=>NOT INTERRUPTIBLE
C2D0:08            355              PHP                  ;SAVE OFLOW
C2D1:20 75 FC      356              JSR   SNIFFIRQ       ;ALLOW IRQ
C2D4:28            357              PLP                  ;RESTORE OFLOW
C2D5:        C2D5  358 IK2A        EQU   *
C2D5:E6 4E         359              INC   RNDL
C2D7:D0 02   C2DB  360              BNE   IK3
C2D9:E6 4F         361              INC   RNDH
C2DB:        C2DB  362 IK3         EQU   *
C2DB:AD 00 C0      363              LDA   KBD            ;KEYPRESS?
C2DE:30 09   C2E9  364              BMI   KDRETY         ;=>YES
C2E0:88            365              DEY
C2E1:D0 EB   C2CE  366              BNE   IK2
C2E3:CA            367              DEX
C2E4:D0 E6   C2CC  368              BNE   IK1
C2E6:        C2E6  369 KDRETN      EQU   *
C2E6:18            370              CLC
C2E7:90 01   C2EA  371              BCC   KDRET
C2E9:        C2E9  372 KDRETY      EQU   *
C2E9:38            373              SEC
C2EA:        C2EA  374 KDRET       EQU   *
C2EA:60            375              RTS
C2EB:              376 *
C2EB:              377 * EXIT. EITHER EXIT WITH OR WITHOUT
C2EB:              378 *  ENABLING I/O SPACE.
C2EB:              379 *
C2EB:        C2EB  380 F.RETURN    EQU   *
C2EB:28            381              PLP                  ;GET PRIOR I/O DISABLE
C2EC:30 03   C2F1  382              BMI   F.RET1         ;=>LEAVE IT DISABLED
C2EE:4C 29 FD      383              JMP   FUNCEXIT       ;=>EXIT & ENABLE I/O
C2F1:4C 2C FD      384 F.RET1      JMP   FUNCEXIT+3     ;EXIT DISABLED
C2F4:              385 ----------------------------------------
C2F4:        C2F4  386 X.CLEOLZ    EQU   *
C2F4:A9 A0         387              LDA   #$A0
C2F6:        C2F6  388 X.CLEOL2    EQU   *
C2F6:91 28         389              STA   (BASL),Y
C2F8:C8            390              INY
C2F9:C4 21         391              CPY   WNDWDTH
C2FB:90 F9   C2F6  392              BCC   X.CLEOL2
C2FD:60            393              RTS                  ;ALWAYS RETURN DIRECTLY
C2FE:        0002  394 ZSPAREC2    EQU   $C300-*
```

```
C2FE:          0002  395              DS    $C300-*,0
C300:                  5              INCLUDE C3SPACE
C300:                  2 ----------------------------------------
C300:                  3 *
C300:                  4 * THIS IS THE $C3XX ROM SPACE:
C300:                  5 *
C300:                  6 ----------------------------------------
C300:          C300    7 CN00         EQU   *
C300:          C300    8 BASICINT     EQU   *
C300: 2C 58 FF         9              BIT   IORTS          ;SET VFLAG (INIT)
C303: 70 12    C317   10              BVS   BASICENT       ;(ALWAYS TAKEN)
C305:          C305   11 BASICIN      EQU   *
C305: 38               12              SEC
C306: 90               13              DFB   $90            ;BCC OPCODE (NEVER TAKEN)
C307:          C307   14 BASICOUT     EQU   *
C307: 18               15              CLC
C308: B8               16              CLV                  ;CLEAR VFLAG (NOT INIT)
C309: 50 0C    C317   17              BVC   BASICENT       ;(ALWAYS TAKEN)
C30B:                  18 *
C30B:                  19 * PASCAL 1.1 FIRMWARE PROTOCOL TABLE:
C30B:                  20 *
C30B: 01               21              DFB   $01            ;GENERIC SIGNATURE BYTE
C30C: 88               22              DFB   $88            ;DEVICE SIGNATURE BYTE
C30D:                  23 *
C30D: 4B               24              DFB   >JPINIT        ;PASCAL INIT
C30E: 51               25              DFB   >JPREAD        ;PASCAL READ
C30F: 57               26              DFB   >JPWRITE       ;PASCAL WRITE
C310: 5D               27              DFB   >JPSTAT        ;PASCAL STATUS
C311:                  28 ----------------------------------------
C311:                  29 *
C311:                  30 * 128K SUPPORT ROUTINE ENTRIES:
C311:                  31 *
C311: 4C 63 C3         32              JMP   MOVE           ;MEMORY MOVE ACROSS BANKS
C314: 4C B0 C3         33              JMP   XFER           ;TRANSFER ACROSS BANKS
C317:                  34 ----------------------------------------
C317:                  35 * BASIC I/O ENTRY POINT:
C317:                  36 ----------------------------------------
C317:          C317    37 BASICENT    EQU   *
C317: 8D 7B 06         38              STA   CHAR           ;SAVE CHARACTER
C31A: 48               39              PHA                  ;SAVE AC
C31B: 98               40              TYA                  ;  AND Y
C31C: 48               41              PHA
C31D: 8A               42              TXA                  ;  AND X
C31E: 48               43              PHA
C31F: 08               44              PHP                  ;SAVE CARRY & VFLAG
C320:                  45 *
C320:                  46 * SET IRQMODE:
C320:                  47 *
C320: AD FB 04         48              LDA   MODE           ;ASSUME IRQ IS DISABLED
C323: 29 FE            49              AND   #255-M.IRQ
C325: 8D FB 04         50              STA   MODE
C328: 68               51              PLA                  ;GET PSTATUS
C329: 48               52              PHA                  ;  AND LEAVE ON STACK
C32A: 29 04            53              AND   #$04           ;IS 'I' BIT SET?
C32C: D0 08    C336    54              BNE   BASICENT2      ;=>YES, DISABLED
C32E: AD FB 04         55              LDA   MODE
C331: 09 01            56              ORA   #M.IRQ
C333: 8D FB 04         57              STA   MODE           ;SET IT ENABLED
C336:          C336    58 BASICENT2   EQU   *
C336: AD FF CF         59              LDA   $CFFF          ;KICK OUT ALL C8 ROMS
C339: A5 25            60              LDA   CV             ;GET USER CV AND
C33B: 8D FB 05         61              STA   OURCV          ;  STUFF IT FOR US
C33E: 20 EB C3         62              JSR   SETC8          ;SETUP C8 INDICATOR
C341: 28               63              PLP                  ;GET VFLAG (INIT)
C342: 08               64              PHP
C343: 70 03    C348    65              BVS   JBASINIT       ;=>DO THE INIT
C345: 4C 66 C8         66              JMP   C8BASIC        ;GET OUT OF CN SPACE
C348: 4C 03 C8         67 JBASINIT    JMP   BASICINIT      ;=>GOTO C8 SPACE
C34B:                  68 *
C34B:          C34B    69 JPINIT      EQU   *
C34B: 20 EB C3         70              JSR   SETC8          ;SETUP C8 INDICATOR
C34E: 4C 4F CA         71              JMP   PINIT          ;XFER TO PASCAL INIT
C351:          C351    72 JPREAD      EQU   *
C351: 20 EB C3         73              JSR   SETC8          ;SETUP C8 INDICATOR
C354: 4C 74 CA         74              JMP   PREAD          ;XFER TO PASCAL READ
C357:          C357    75 JPWRITE     EQU   *
C357: 20 EB C3         76              JSR   SETC8          ;SETUP C8 INDICATOR
C35A: 4C 8E CA         77              JMP   PWRITE         ;XFER TO PASCAL WRITE
C35D:          C35D    78 JPSTAT      EQU   *
C35D: 20 EB C3         79              JSR   SETC8          ;SETUP C8 INDICATOR
C360: 4C 94 C9         80              JMP   PSTATUS        ;XFER TO PASCAL STATUS
C363:                  82 ----------------------------------------
C363:                  83 * NAME     : MOVE
C363:                  84 * FUNCTION: PERFORM CROSSBANK MEMORY MOVE
C363:                  85 * INPUT    : A1=SOURCE ADDRESS
C363:                  86 *          : A2=SOURCE END
C363:                  87 *          : A4=DESTINATION START
C363:                  88 *          : CARRY SET=MAIN-->CARD
C363:                  89 *                CLR=CARD-->MAIN
C363:                  90 * OUTPUT   : NONE
C363:                  91 * VOLATILE: NOTHING
```

```
C363:                    92 * CALLS    : NOTHING
C363:                    93 ----------------------------------------
C363:                    94 *
C363:          C363      95 MOVE         EQU   *
C363: 48                 96               PHA                  ;SAVE AC
C364: 98                 97               TYA                  ; AND Y
C365: 48                 98               PHA
C366: AD 13 C0           99               LDA   RDRAMRD        ;SAVE STATE OF
C369: 48                100               PHA                  ; MEMORY FLAGS
C36A: AD 14 C0          101               LDA   RDRAMWRT
C36D: 48                102               PHA
C36E:                   103 *
C36E:                   104 * SET FLAGS FOR CROSSBANK MOVE:
C36E:                   105 *
C36E: 90 08    C378     106               BCC   MOVEC2M        ;=>CARD-->MAIN
C370: 8D 02 C0          107               STA   RDMAINRAM      ;SET FOR MAIN
C373: 8D 05 C0          108               STA   WRCARDRAM      ; TO CARD
C376: B0 06    C37E     109               BCS   MOVESTRT       ;=>(ALWAYS TAKEN)
C378:                   110 *
C378:          C378     111 MOVEC2M     EQU   *
C378: 8D 04 C0          112               STA   WRMAINRAM      ;SET FOR CARD
C37B: 8D 03 C0          113               STA   RDCARDRAM      ; TO MAIN
C37E:                   114 *
C37E:          C37E     115 MOVESTRT    EQU   *
C37E: A0 00             116               LDY   #0             ;DUMMY INDEX
C380:                   117 *
C380:          C380     118 MOVELOOP    EQU   *
C380: B1 3C             119               LDA   (A1L),Y        ;GET A BYTE
C382: 91 42             120               STA   (A4L),Y        ;MOVE IT
C384: E6 42             121               INC   A4L
C386: D0 02    C38A     122               BNE   NXTA1
C388: E6 43             123               INC   A4H
C38A: A5 3C             124 NXTA1       LDA   A1L
C38C: C5 3E             125               CMP   A2L
C38E: A5 3D             126               LDA   A1H
C390: E5 3F             127               SBC   A2H
C392: E6 3C             128               INC   A1L
C394: D0 02    C398     129               BNE   C01
C396: E6 3D             130               INC   A1H
C398: 90 E6    C380     131 C01           BCC   MOVELOOP       ;=>MORE TO MOVE
C39A:                   132 *
C39A:                   133 * RESTORE ORIGINAL FLAGS:
C39A:                   134 *
C39A: 8D 04 C0          135               STA   WRMAINRAM      ;CLEAR FLAG2
C39D: 68                136               PLA                  ;GET ORIGINAL STATE
C39E: 10 03    C3A3     137               BPL   C03            ;=>IT WAS OFF
C3A0: 8D 05 C0          138               STA   WRCARDRAM
C3A3:          C3A3     139 C03           EQU   *
C3A3: 8D 02 C0          140               STA   RDMAINRAM      ;CLEAR FLAG1
C3A6: 68                141               PLA                  ;GET ORIGINAL STATE
C3A7: 10 03    C3AC     142               BPL   MOVERET        ;=>IT WAS OFF
C3A9: 8D 03 C0          143               STA   RDCARDRAM
C3AC:          C3AC     144 MOVERET     EQU   *
C3AC: 68                145               PLA                  ;RESTORE Y
C3AD: A8                146               TAY
C3AE: 68                147               PLA                  ; AND AC
C3AF: 60                148               RTS
C3B0:                   149 ----------------------------------------
C3B0:                   150 * NAME     : XFER
C3B0:                   151 * FUNCTION: TRANSFER CONTROL CROSSBANK
C3B0:                   152 * INPUT    : $03ED=TRANSFER ADDR
C3B0:                   153 *          : CARRY SET=XFER TO CARD
C3B0:                   154 *                CLR=XFER TO MAIN
C3B0:                   155 *          : VFLAG CLR=USE STD ZP/STK
C3B0:                   156 *          :       SET=USE ALT ZP/STK
C3B0:                   157 * OUTPUT   : NONE
C3B0:                   158 * VOLATILE: $03ED/03EE IN DEST BANK
C3B0:                   159 * CALLS    : NOTHING
C3B0:                   160 * NOTE     : ENTERED VIA JMP, NOT JSR
C3B0:                   161 ----------------------------------------
C3B0:                   162 *
C3B0:          C3B0     163 XFER        EQU   *
C3B0: 48                164               PHA                  ;SAVE AC ON CURRENT STACK
C3B1:                   165 *
C3B1:                   166 * COPY DESTINATION ADDRESS TO THE
C3B1:                   167 *  OTHER BANK SO THAT WE HAVE IT
C3B1:                   168 *  IN CASE WE DO A SWAP:
C3B1:                   169 *
C3B1: AD ED 03          170               LDA   $03ED          ;GET XFERADDR LO
C3B4: 48                171               PHA                  ;SAVE ON CURRENT STACK
C3B5: AD EE 03          172               LDA   $03EE          ;GET XFERADDR HI
C3B8: 48                173               PHA                  ;SAVE IT TOO
C3B9:                   174 *
C3B9:                   175 * SWITCH TO APPROPRIATE BANK:
C3B9:                   176 *
C3B9: 90 0A    C3C5     177               BCC   XFERC2M        ;=>CARD-->MAIN
C3BB: 8D 03 C0          178               STA   RDCARDRAM      ;SET FOR RUNNING
C3BE: 8D 05 C0          179               STA   WRCARDRAM      ; IN CARD RAM
C3C1: 50 19    C3DC     180               BVC   XFERSZP        ;=>USE STD ZP/STK
C3C3: 70 08    C3CD     181               BVS   XFERAZP        ;=>USE ALT ZP/STK
C3C5:          C3C5     182 XFERC2M     EQU   *
```

```
C3C5:8D 02 C0      183              STA   RDMAINRAM   ;SET FOR RUNNING
C3C8:8D 04 C0      184              STA   WRMAINRAM   ; IN MAIN RAM
C3CB:50 0F   C3DC  185              BVC   XFERSZP     ;=>USE STD ZP/STK
C3CD:              186 *
C3CD:        C3CD  187 XFERAZP      EQU   *           ;SWITCH TO ALT ZP/STK
C3CD:68            188              PLA               ;STUFF XFERADDR
C3CE:8D EE 03      189              STA   $03EE       ; HI AND
C3D1:68            190              PLA
C3D2:8D ED 03      191              STA   $03ED       ;  LO
C3D5:68            192              PLA               ;RESTORE AC
C3D6:8D 09 C0      193              STA   SETALTZP    ;SWITCH TO ALT ZP/STK
C3D9:6C ED 03      194              JMP   ($03ED)     ;=>OFF WE GO!
C3DC:              195 *
C3DC:        C3DC  196 XFERSZP      EQU   *
C3DC:68            197              PLA               ;STUFF XFERADDR
C3DD:8D EE 03      198              STA   $03EE       ; HI AND
C3E0:68            199              PLA
C3E1:8D ED 03      200              STA   $03ED       ;  LO
C3E4:68            201              PLA               ;RESTORE AC
C3E5:8D 08 C0      202              STA   SETSTDZP    ;=>SWITCH TO STD ZP/STK
C3E8:6C ED 03      203              JMP   ($03ED)     ;OFF WE GO!
C3EB:              204 ----------------------------------------
C3EB:              205 * NAME     : SETC8
C3EB:              206 * FUNCTION: SETUP IRQ $C800 PROTOCOL
C3EB:              207 * INPUT    : NONE
C3EB:              208 * OUTPUT   : NONE
C3EB:              209 * VOLATILE: NOTHING
C3EB:              210 * CALLS    : NOTHING
C3EB:              211 ----------------------------------------
C3EB:              212 *
C3EB:        C3EB  213 SETC8        EQU   *
C3EB:48            214              PHA               ;SAVE AC
C3EC:A9 C3         215              LDA   #<CN00      ;SLOT NUMBER
C3EE:8D F8 07      216              STA   C8SLOT      ;STUFF IT
C3F1:68            217              PLA               ;RESTORE AC
C3F2:60            218              RTS
C3F3:                6              INCLUDE C8SPACE
C3F3:                2 ----------------------------------------
C3F3:                3 * THIS IS THE C8XX SPACE:
C3F3:                4 ----------------------------------------
C3F3:        0000    5              DO    TEST
 S                   6              ORG   $D800
C3F3:                7              ELSE
----- NEXT OBJECT FILE NAME IS VIDEO.OBJ1
C800:        C800    8              ORG   $C800
C800:                9              FIN
C800:4C 4A CA       10              JMP   PINIT1.0    ;PASCAL 1.0 INIT
C803:               11 * BASIC INITIALIZATION:
C803:               12 ----------------------------------------
C803:        C803   13 BASICINIT    EQU   *
C803:A9 06          14              LDA   #GOODF8     ;CHECK F8 ROM
C805:CD B3 FB       15              CMP   F8VERSION   ; IS IT OK?
C808:F0 0C   C816   16              BEQ   BINIT1      ;=>YES
C80A:20 78 CF       17              JSR   COPYROM     ;TRY COPYING TO RAMCARD
C80D:CD B3 FB       18              CMP   F8VERSION
C810:F0 04   C816   19              BEQ   BINIT1      ;=>NOW IT'S GOOD
C812:78             20              SEI               ;CRASH THE SYSTEM!
C813:        C813   21 HANG         EQU   *
C813:4C 13 C8       22              JMP   HANG        ;HANG FOREVER
C816:               23 *
C816:        C816   24 BINIT1       EQU   *
C816:A9 C3          25              LDA   #<CN00      ;SET HOOKS FOR
C818:85 37          26              STA   CSWH
C81A:85 39          27              STA   KSWH        ; IN & OUT
C81C:A9 05          28              LDA   #>BASICIN
C81E:85 38          29              STA   KSWL
C820:A9 07          30              LDA   #>BASICOUT
C822:85 36          31              STA   CSWL
C824:A9 00          32              LDA   #0          ;SET FULL 40-COL WINDOW
C826:85 20          33              STA   WNDLFT
C828:A9 00          34              LDA   #0          ;ASSUME TEXT MODE
C82A:2C 1A C0       35              BIT   RDTEXT      ;IN TEXT MODE?
C82D:30 02   C831   36              BMI   BINIT1A     ;=>YES
C82F:A9 14          37              LDA   #20         ;IF GR, SET 4 LINES
C831:        C831   38 BINIT1A      EQU   *
C831:85 22          39              STA   WNDTOP
C833:A9 18          40              LDA   #24
C835:85 23          41              STA   WNDBTM
C837:A9 28          42              LDA   #40
C839:85 21          43              STA   WNDWDTH
C83B:A5 24          44              LDA   CH          ;COPY USER CH
C83D:8D 7B 04       45              STA   OLDCH       ; AS 'OLD' SETTING
C840:A9 01          46              LDA   #M.IRQ      ;GET READY TO CLEAR
C842:2D FB 04       47              AND   MODE        ;PRESERVE IRQ STATUS
C845:8D FB 04       48              STA   MODE        ;CLEAR MODES
C848:4C 50 C8       49              JMP   BINIT2      ;=>CONTINUE AFTER PASCAL 1.0 HOOK
C84B:               50 ----------------------------------------
C84B:               51 *
C84B:               52 * PASCAL 1.0 INPUT HOOK:
C84B:               53 *
C84B:00             54              BRK
```

```
C84C: 00              55              BRK
C84D:           0000  56              IFNE  *-$C84D     ;ERR IF WRONG ADDR
 S                    57              FAIL  2,'C84D     HOOK ALIGNMENT'
C84D:                 58              FIN
C84D: 4C 51 C3        59              JMP   JPREAD      ;=>GO TO STANDARD READ
C850:                 60 ---------------------------------------
C850:                 61 *
C850:                 62 * IS THERE A CARD?
C850:                 63 *
C850:           C850  64 BINIT2       EQU   *
C850: 20 24 CB        65              JSR   TESTCARD    ;SEE IF CARD PLUGGED IN
C853: D0 08     C85D  66              BNE   CLEARIT     ;=>IT'S 40
C855: 06 21           67              ASL   WNDWDTH     ;SET 80-COL WINDOW
C857: 8D 01 C0        68              STA   SET80COL    ;ENABLE 80 STORE
C85A: 8D 0D C0        69              STA   SET80VID    ; AND 80 VIDEO
C85D:                 70 *
C85D:                 71 * HOME & CLEAR:
C85D:                 72 *
C85D:           C85D  73 CLEARIT      EQU   *
C85D: 8D 0F C0        74              STA   SETALTCHAR  ;SET NORM/INV LCASE
C860: 20 42 CD        75              JSR   X.FF        ;CLEAR IT
C863: 28              76              PLP               ;CLC ASSURES THAT
C864: 18              77              CLC               ; WE PRINT THIS
C865: 08              78              PHP               ;  INITIAL CHARACTER
C866:                 80 *
C866:                 81 * COMPENSATE FOR INTEGER BASIC'S
C866:                 82 *  HITTING OF $C000 ON INITIAL ENTRY:
C866:                 83 *
C866:           C866  84 C8BASIC      EQU   *           ;BASIC IN/OUT
C866: 2C 1F C0        85              BIT   RD80VID     ;WHICH MODE?
C869: 10 09     C874  86              BPL   C8B2        ;=>40.  LEAVE ALONE
C86B: 8D 01 C0        87              STA   SET80COL    ;80.  ENABLE STORE.
C86E:                 88 *
C86E:                 89 * MAKE SURE SCROLLING WINDOW IS
C86E:                 90 *  AN EVEN NUMBER FOR 80-COLS:
C86E:                 91 *
C86E: A5 21           92              LDA   WNDWDTH
C870: 29 FE           93              AND   #$FE
C872: 85 21           94              STA   WNDWDTH     ;ROUND IT TO LOWER EVEN
C874:                 95 *
C874:                 96 * COPY USER'S CH IF IT DIFFERS FROM
C874:                 97 *  WHAT WE LAST PUT THERE:
C874:                 98 *
C874:           C874  99 C8B2         EQU   *
C874: A5 24          100              LDA   CH          ;GET IT
C876: CD 7B 04       101              CMP   OLDCH       ;IS IT THE SAME?
C879: F0 03     C87E 102              BEQ   C8B3        ;=>YES, USE OUR OWN
C87B: 8D 7B 05       103              STA   OURCH       ;=>NO,  USE HIS
C87E:           C87E 104 C8B3         EQU   *
C87E: A9 06          105              LDA   #GOODFB     ;CHECK F8 ROM
C880: CD B3 FB       106              CMP   F8VERSION   ;IF DIFFERENT, USER
C883: F0 0B     C890 107              BEQ   C8B4        ; HAS RELOADED RAMCARD
C885:                108 *
C885:                109 * COPY F8 ROM TO LANG CARD:
C885:                110 *
C885: 20 78 CF       111              JSR   COPYROM     ;COPY IT AGAIN
C888: CD B3 FB       112              CMP   F8VERSION   ;IS IT NOW CORRECT?
C88B: F0 03     C890 113              BEQ   C8B4        ;=>GREAT
C88D: 4C 13 C8       114              JMP   HANG        ;=>WE HAVE WRONG ROM!
C890:                115 *
C890:           C890 116 C8B4         EQU   *
C890: 28             117              PLP               ;RECOVER CARRY (IN/OUT)
C891: 90 03     C896 118              BCC   BOUT        ;=>PRINT A CHAR
C893: 4C F6 C8       119              JMP   BINPUT      ;=>INPUT A CHAR
C896:           C896 120 BOUT         EQU   *
C896: AD FB 04       121              LDA   MODE        ;SAY THAT WE'RE
C899: 29 BF          122              AND   #255-M.BINPUT ; PRINTING
C89B: 8D FB 04       123              STA   MODE
C89E: 4C A1 C8       124              JMP   BPRINT      ;->OUTPUT A CHAR
C8A1:                  7              INCLUDE BPRINT
C8A1:                  2 ---------------------------------------
C8A1:                  3 * BASIC OUTPUT:
C8A1:                  4 ---------------------------------------
C8A1:           C8A1   5 BPRINT       EQU   *
C8A1: AD 7B 06         6              LDA   CHAR        ;GET CHARACTER
C8A4: C9 8D            7              CMP   #$8D        ;IS IT C/R?
C8A6: D0 18     C8C0   8              BNE   NOWAIT      ;=>NOPE, NO VIDWAIT
C8A8: AC 00 C0         9              LDY   KBD         ;IS KEY PRESSED?
C8AB: 10 13     C8C0  10              BPL   NOWAIT      ;NO
C8AD: C0 93           11              CPY   #$93        ;IS IT CTL-S?
C8AF: D0 0F     C8C0  12              BNE   NOWAIT      ;NO, IGNORE IT
C8B1: 2C 10 C0        13              BIT   KBDSTRB     ;CLEAR STROBE
C8B4: AC 00 C0        14 KBDWAIT      LDY   KBD         ;WAIT FOR NEXT KEYPRESS
C8B7: 10 FB     C8B4  15              BPL   KBDWAIT
C8B9: C0 83           16              CPY   #$83        ;IF CTL-C, LEAVE IT
C8BB: F0 03     C8C0  17              BEQ   NOWAIT      ;  IN THE KBD BUFFER
C8BD: 2C 10 C0        18              BIT   KBDSTRB     ;CLEAR OTHER CHARACTER
C8C0:           C8C0  19 NOWAIT       EQU   *
C8C0: 29 7F           20              AND   #$7F        ;DROP POSSIBLE HI BIT
C8C2: C9 20           21              CMP   #$20        ;IS IT CONTROL CHAR?
C8C4: B0 06     C8CC  22              BCS   BPNCTL      ;=>NOPE
```

```
C8C6:20 99 CB       23              JSR   CTLCHAR     ;EXECUTE POSSIBLE CTL CHAR
C8C9:4C E2 C8       24              JMP   BIORET      ;=>EXECUTED OR IGNORED
C8CC:               25 *
C8CC:               26 * NOT A CTL CHAR. PRINT IT.
C8CC:               27 *
C8CC:         C8CC  28 BPNCTL       EQU   *
C8CC:AC 7B 05       29              LDY   OURCH       ;GET CH
C8CF:AD 7B 06       30              LDA   CHAR        ;GET CHAR (ALL 8 BITS)
C8D2:20 F2 CE       31              JSR   STORCHAR    ;STUFF ONTO SCREEN
C8D5:               32 *
C8D5:               33 * BUMP THE CURSOR HORIZONTAL:
C8D5:               34 *
C8D5:EE 7B 05       35              INC   OURCH       ;BUMP IT
C8D8:AD 7B 05       36              LDA   OURCH       ;ARE WE PAST THE
C8DB:C5 21          37              CMP   WNDWDTH     ; END OF THE LINE?
C8DD:90 03    C8E2  38              BCC   BIORET      ;=>NO, NO PROBLEM
C8DF:20 EC CB       39              JSR   X.CR        ;YES, DO C/R
C8E2:               40 *
C8E2:         C8E2  41 BIORET       EQU   *
C8E2:AD 7B 05       42              LDA   OURCH       ;SET CH AND CV
C8E5:20 AF CE       43              JSR   SETCH       ; FOR BASIC
C8E8:AD FB 05       44              LDA   OURCV
C8EB:85 25          45              STA   CV
C8ED:68             46              PLA               ;RESTORE
C8EE:AA             47              TAX
C8EF:68             48              PLA               ;X AND Y
C8F0:A8             49              TAY
C8F1:68             50              PLA               ; AND AC
C8F2:AD 7B 06       51              LDA   CHAR
C8F5:60             52              RTS               ;RETURN TO BASIC
C8F6:                8              INCLUDE BINPUT
C8F6:                2 * BASIC INPUT:
C8F6:                3 *
C8F6:         C8F6   4 BINPUT       EQU   *
C8F6:AD FB 04        5              LDA   MODE        ;SAY THAT
C8F9:09 40           6              ORA   #M.BINPUT   ; WE'RE INPUTTING
C8FB:8D FB 04        7              STA   MODE
C8FE:AD 7B 06        8              LDA   CHAR        ;GET CHAR AT CURSOR AND
C901:A4 24           9              LDY   CH          ;GET CURSOR POSITION
C903:91 28          10              STA   (BASL),Y    ; REPAIR MONITOR'S SILLY ATTEMPT
C905:         C905  11 B.INPUT      EQU   *
C905:20 DD CE       12              JSR   INVERT      ;CREATE OUR OWN CURSOR IMAGE
C908:20 15 CB       13              JSR   GETKEY      ;GET A KEY
C90B:8D 7B 06       14              STA   CHAR        ;SAVE IT
C90E:20 DD CE       15              JSR   INVERT      ;REMOVE CURSOR
C911:C9 9B          16              CMP   #$9B        ;ESCAPE KEY?
C913:F0 03    C918  17              BEQ   ESCAPING    ;=>YES IT IS
C915:4C B7 C9       18              JMP   NOESC       ;=>NO, IT'S NORMAL
C918:               20 * START AN ESCAPE SEQUENCE:
C918:               21 *  WE HANDLE THE FOLLOWING ONES:
C918:               22 *     @ - HOME & CLEAR
C918:               23 *     E - CLR TO EOL
C918:               24 *     F - CLR TO EOS
C918:               25 *     I - CURSOR UP
C918:               26 *     J - CURSOR LEFT
C918:               27 *     K - CURSOR RIGHT
C918:               28 *     M - CURSOR DOWN
C918:               29 *     R - RESTRICT TO UPPERCASE
C918:               30 *     T - TURN OFF ESC-R
C918:               31 *     4 - GOTO 40 COLUMN MODE
C918:               32 *     8 - GOTO 80 COLUMN MODE
C918:               33 * CTL-G- QUIT (PR#0/IN#0)
C918:               34 *    THE FOUR ARROW KEYS (AS IJKM)
C918:               35 *
C918:               36              MSB   OFF
C918:         C918  37 ESCAPING     EQU   *
C918:20 52 CF       38              JSR   ESCON       ;ESCAPE CURSORON
C91B:20 15 CB       39              JSR   GETKEY      ;GET ESCAPE FUNCTION
C91E:20 65 CF       40              JSR   ESCOFF      ;REPLACE ORIGINAL CHARACTER
C921:29 7F          41              AND   #$7F        ;DROP HI BIT
C923:C9 60          42              CMP   #$60        ;IS IT LOWERCASE?
C925:90 02    C929  43              BCC   ESC1        ;=>NO, DON'T UPSHIFT
C927:29 DF          44              AND   #255-$20    ;UPSHIFT
C929:         C929  45 ESC1         EQU   *
C929:A0 11          46              LDY   #ESCNUM     ;COUNT/INDEX
C92B:         C92B  47 ESC2         EQU   *
C92B:D9 72 C9       48              CMP   ESCTAB,Y    ;IS IT A VALID ESCAPE?
C92E:F0 05    C935  49              BEQ   ESC3        ;=>YES
C930:88             50              DEY
C931:10 F8    C92B  51              BPL   ESC2        ;TRY 'EM ALL...
C933:30 10    C945  52              BMI   ESCSPEC     ;=>MAYBE IT'S A SPECIAL ONE
C935:               53 *
C935:         C935  54 ESC3         EQU   *
C935:B9 83 C9       55              LDA   ESCCHAR,Y   ;GET CHAR TO "PRINT"
C938:29 7F          56              AND   #$7F        ;DROP HI BIT (FLAG)
C93A:20 99 CB       57              JSR   CTLCHAR     ;EXECUTE IT
C93D:B9 83 C9       58              LDA   ESCCHAR,Y   ;GET FLAG
C940:30 D6    C918  59              BMI   ESCAPING    ;=>STAY IN ESCAPE MODE
C942:4C 05 C9       60              JMP   B.INPUT     ;=>QUIT ESCAPE MODE
C945:               61 *
C945:         C945  62 ESCSPEC      EQU   *
```

```
C945: C9 11          63              CMP   #$11          ;IS IT ESC-CTLQ?
C947: D0 0B    C954  64              BNE   ESCSPEC2      ;=>NO
C949: 20 AA CD       65              JSR   QUIT          ;DO THE QUITTING STUFF
C94C: A9 98          66              LDA   #$98          ;RETURN CTL-X AS
C94E: 8D 7B 06       67              STA   CHAR          ;  THE CHARACTER
C951: 4C E2 C8       68              JMP   BIORET        ;=>QUIT THE CARD FOREVER
C954:                69 *
C954:          C954  70 ESCSPEC2     EQU   *
C954: C9 52          71              CMP   #'R'          ;IS IT ESC-R?
C956: D0 0B    C963  72              BNE   ESCSPEC3      ;=>NO
C958: AD FB 04       73              LDA   MODE          ;YES, SET IT
C95B: 09 80          74              ORA   #M.ESCR
C95D: 8D FB 04       75              STA   MODE
C960: 4C 05 C9       76 ESCNONE      JMP   B.INPUT       ;QUIT ESCAPE MODE
C963:                77 *
C963:          C963  78 ESCSPEC3     EQU   *
C963: C9 54          79              CMP   #'T'          ;IS IT ESC-T?
C965: D0 F9    C960  80              BNE   ESCNONE       ;=>NOTHING
C967: AD FB 04       81              LDA   MODE
C96A: 29 7F          82              AND   #255-M.ESCR
C96C: 8D FB 04       83              STA   MODE
C96F: 4C 05 C9       84              JMP   B.INPUT       ;QUIT ESCAPE MODE
C972:          C972  86 ESCTAB       EQU   *
C972: 40             87              ASC   '@'
C973: 41             88              ASC   'A'           ;HANDLE OLD ESCAPES
C974: 42             89              ASC   'B'
C975: 43             90              ASC   'C'
C976: 44             91              ASC   'D'
C977: 45             92              ASC   'E'
C978: 46             93              ASC   'F'
C979: 49             94              ASC   'I'
C97A: 4A             95              ASC   'J'
C97B: 4B             96              ASC   'K'
C97C: 4D             97              ASC   'M'
C97D: 34             98              ASC   '4'
C97E: 38             99              ASC   '8'
C97F: 08            100              DFB   $08           ;LEFT ARROW
C980: 0A            101              DFB   $0A           ;DOWN ARROW
C981: 0B            102              DFB   $0B           ;UP   ARROW
C982: 15            103              DFB   $15           ;RITE ARROW
C983:          0011 104 ESCNUM       EQU   *-ESCTAB
C983:                105              MSB   ON
C983:          C983 106 ESCCHAR      EQU   *
C983: 0C            107              DFB   $0C+$00       ;@: FORMFEED
C984: 1C            108              DFB   $1C           ;A: FS
C985: 08            109              DFB   $08           ;B: BS
C986: 0A            110              DFB   $0A           ;C: LF
C987: 1F            111              DFB   $1F           ;D: US
C988: 1D            112              DFB   $1D+$00       ;E: GS
C989: 0B            113              DFB   $0B+$00       ;F: VT
C98A: 9F            114              DFB   $1F+$80       ;I: US (STAY ESC)
C98B: 88            115              DFB   $0B+$80       ;J: BS (STAY ESC)
C98C: 9C            116              DFB   $1C+$80       ;K: FS (STAY ESC)
C98D: 8A            117              DFB   $0A+$80       ;M: LF (STAY ESC)
C98E: 11            118              DFB   $11+$00       ;4 :DC1
C98F: 12            119              DFB   $12+$00       ;8 :DC2
C990: 88            120              DFB   $0B+$80       ;<-:BS (STAY ESC)
C991: 8A            121              DFB   $0A+$80       ;DN:LF (STAY ESC)
C992: 9F            122              DFB   $1F+$80       ;UP:US (STAY ESC)
C993: 9C            123              DFB   $1C+$80       ;->:FS (STAY ESC)
C994:               124 ----------------------------------------
C994:               126 ----------------------------------------
C994:               127 * PASCAL STATUS:
C994:               128 ----------------------------------------
C994:          C994 129 PSTATUS      EQU   *
C994: AA            130              TAX                 ;SAVE REQUEST CODE
C995: 20 C8 CF      131              JSR   PSETUP        ;SETUP ZP STUFF
C998: 8A            132              TXA                 ;IS IT 'READY FOR OUTPUT?'
C999: D0 03    C99E 133              BNE   PSTATUS2      ;=>NO
C99B: 38            134              SEC                 ;YES, READY FOR OUTPUT
C99C: B0 16    C9B4 135              BCS   PSTATUS4
C99E:               136 *
C99E:          C99E 137 PSTATUS2     EQU   *
C99E: C9 01         138              CMP   #1            ;IS IT 'ANY INPUT?'
C9A0: F0 0E    C9B0 139              BEQ   PSTATUS3      ;=>YES
C9A2: A2 03         140              LDX   #3            ;IORESULT='ILGL OPERATION'
C9A4: 18            141              CLC
C9A5: 60            142              RTS
C9A6:               143 ----------------------------------------
C9A6:               144 * PASCAL 1.0 OUTPUT HOOK:
C9A6:               145 ----------------------------------------
C9A6: 00            146              BRK                 ;PADDING
C9A7: 00            147              BRK
C9A8: 00            148              BRK
C9A9: 00            149              BRK
C9AA:          0000 150              IFNE  *-$C9AA
 S                  151              FAIL  2,'C9AA       HOOK ALIGNMENT'
C9AA:               152              FIN
C9AA: AD 7B 06      153              LDA   CHAR          ;GET OUTPUT CHARACTER
C9AD: 4C 57 C3      154              JMP   JPWRITE       ;=>USE STANDARD WRITE
C9B0:               155 *
```

```
C9B0:             C9B0  156 PSTATUS3    EQU   *
C9B0: AD 00 C0        157               LDA   KBD          ;IS THERE A KEYPRESS?
C9B3: 0A              158               ASL   A            ;STROBE-->CARRY
C9B4: A2 00           159 PSTATUS4      LDX   #0           ;IORESULT='GOOD'
C9B6: 60              160               RTS
C9B7:                 162 ----------------------------------------
C9B7:                 163 *---- BASIC INPUT, CONTINUED:
C9B7:                 164 *---- NOT AN ESCAPE SEQUENCE----
C9B7:                 165 ----------------------------------------
C9B7:             C9B7  166 NOESC         EQU   *            ;NOT ESCAPE KEY
C9B7:                 167 *
C9B7: C9 95           168               CMP   #$95         ;IS IT PICK?
C9B9: D0 0B       C9C6  169               BNE   B.NOPICK     ;=>NOPE
C9BB: AC 7B 05        170               LDY   OURCH        ;YOU CAN PICK YER FRIENDS....
C9BE: 20 01 CF        171               JSR   PICK         ;YES, PICK THE CHAR
C9C1: 09 80           172               ORA   #$80         ;ALWAYS PICK AS NORMAL
C9C3: 8D 7B 06        173               STA   CHAR         ;SAVE AS KEYSTROKE
C9C6:                 174 *
C9C6:                 175 * TRACK QUOTATION MARKS FOR THE
C9C6:                 176 *  RESTRICT-UPPERCASE FEATURE:
C9C6:                 177 *
C9C6:             C9C6  178 B.NOPICK      EQU   *
C9C6: AD FB 04        179               LDA   MODE         ;ARE WE DOING LITERAL INPUT?
C9C9: 29 10           180               AND   #M.LIT
C9CB: D0 12       C9DF  181               BNE   B.CHKCAN     ;=>YES
C9CD:                 182 *
C9CD:                 183 * LITERAL INPUT'S INACTIVE.  SEE IF
C9CD:                 184 *  WE CAN START LITERAL INPUT:
C9CD:                 185 *
C9CD: AD 7B 06        186               LDA   CHAR         ;GET THE CHAR
C9D0: C9 A2           187               CMP   #$A2         ;IS IT A DOUBLE QUOTE?
C9D2: F0 23       C9F7  188               BEQ   B.FLIP       ;=>YES, FLIP LITERAL MODE
C9D4: C9 88           189               CMP   #$88         ;IS HE MOVING LEFT?
C9D6: D0 32       CA0A  190               BNE   B.FIXCHR     ;=>NOPE, JUST REG CHAR
C9D8: 20 27 CA        191               JSR   GETPRIOR     ;GRAB PRIOR CHAR
C9DB: D0 2D       CA0A  192               BNE   B.FIXCHR     ;=>NOT DELETING A QUOTE
C9DD: F0 18       C9F7  193               BEQ   B.FLIP       ;(ALWAYS) HE'S DELETED THE QUOTE
C9DF:                 194 *
C9DF:                 195 * LITERAL INPUT'S ACTIVE.  SEE IF
C9DF:                 196 *  IT SHOULD BE CANCELLED YET:
C9DF:                 197 *
C9DF:             C9DF  198 B.CHKCAN      EQU   *
C9DF: AD 7B 06        199               LDA   CHAR         ;GET CURRENT CHAR
C9E2: C9 A2           200               CMP   #$A2         ;IS CURRENT CHAR THE CLOSING QUOTE?
C9E4: F0 1C       CA02  201               BEQ   B.CANLIT     ;=>YES
C9E6: C9 98           202               CMP   #$98         ;CANCEL LITERAL INPUT
C9E8: F0 18       CA02  203               BEQ   B.CANLIT     ;  IF CTLX OR RETURN
C9EA: C9 8D           204               CMP   #$8D         ;   OR BACK OVER "
C9EC: F0 14       CA02  205               BEQ   B.CANLIT
C9EE: C9 88           206               CMP   #$88         ;BACKSPACE?
C9F0: D0 18       CA0A  207               BNE   B.FIXCHR     ;=>NO, NOT DELETING QUOTE
C9F2: 20 27 CA        208               JSR   GETPRIOR     ;GET CHAR HE'S DELETING
C9F5: D0 13       CA0A  209               BNE   B.FIXCHR     ;=>NOT DELETING A QUOTE
C9F7:                 210 *
C9F7:             C9F7  211 B.FLIP        EQU   *
C9F7: AD FB 04        212               LDA   MODE         ;FLIP THE MODE
C9FA: 49 10           213               EOR   #M.LIT
C9FC: 8D FB 04        214               STA   MODE
C9FF: 4C 0A CA        215               JMP   B.FIXCHR
CA02:             CA02  216 B.CANLIT      EQU   *
CA02: AD FB 04        217               LDA   MODE
CA05: 29 EF           218               AND   #255-M.LIT   ;CANCEL LITERAL INPUT
CA07: 8D FB 04        219               STA   MODE
CA0A:                 220 *
CA0A:             CA0A  221 B.FIXCHR      EQU   *
CA0A: AD FB 04        222               LDA   MODE         ;ESC-R FACILITY ACTIVE?
CA0D: 29 80           223               AND   #M.ESCR
CA0F: F0 13       CA24  224               BEQ   B.INRET      ;=>NOPE
CA11: AD FB 04        225               LDA   MODE         ;LITERAL INPUT ACTIVE?
CA14: 29 10           226               AND   #M.LIT
CA16: D0 0C       CA24  227               BNE   B.INRET      ;=>YES, NO UPSHIFT
CA18: AD 7B 06        228               LDA   CHAR         ;GET THE CHAR
CA1B: C9 E0           229               CMP   #$E0         ;IS CHAR LOWERCASE?
CA1D: 90 05       CA24  230               BCC   B.INRET      ;=>NO, NO NEED TO SHIFT IT
CA1F: 29 DF           231               AND   #$DF         ;RESTRICT TO U/C
CA21: 8D 7B 06        232               STA   CHAR
CA24:                 233 ----------------------------------------
CA24:             CA24  234 B.INRET       EQU   *
CA24: 4C E2 C8        235               JMP   BIORET       ;=>RETURN TO CALLER
CA27:                 237 ----------------------------------------
CA27:                 238 * NAME     : GETPRIOR
CA27:                 239 * FUNCTION: GET CHAR BEFORE CURSOR
CA27:                 240 * INPUT    : OURCH,OURCV
CA27:                 241 * OUTPUT   : 'BEQ' IF CHAR=DBL QUOTE
CA27:                 242 *          : 'BNE' IF NOT
CA27:                 243 * VOLATILE: AC,'TEMP1'
CA27:                 244 * CALLS    : PICK,X.BS,X.FS
CA27:                 245 ----------------------------------------
CA27:                 246 *
CA27:             CA27  247 GETPRIOR      EQU   *
CA27: AD FB 05        248               LDA   OURCV        ;DON'T TRY TO LOOK
```

```
CA2A:0D 7B 05      249              ORA   OURCH        ; BACK IF @ UPPER-LEFT
CA2D:F0 1A   CA49  250              BEQ   GPX          ;  CORNER OF WINDOW!!!
CA2F:98            251              TYA                ;SAVE Y
CA30:48            252              PHA
CA31:20 DB CB      253              JSR   X.BS         ;BACK UP 1 CHAR
CA34:AC 7B 05      254              LDY   OURCH        ;GET CH AND
CA37:20 01 CF      255              JSR   PICK         ;  PICK PRIOR CHAR
CA3A:09 80         256              ORA   #$80         ;PICK AS NORMAL VIDEO
CA3C:8D 7B 04      257              STA   TEMP1        ;HOLD CHAR
CA3F:20 26 CC      258              JSR   X.FS
CA42:68            259              PLA                ;RESTORE
CA43:A8            260              TAY                ; Y
CA44:AD 7B 04      261              LDA   TEMP1
CA47:C9 A2         262              CMP   #$A2         ;IS IT DBL QUOTE?
CA49:              CA49  263 GPX    EQU   *
CA49:60            264              RTS                ;RETURN WITH BEQ/BNE
CA4A:                 9             INCLUDE PINIT
CA4A:                 2 ----------------------------------------
CA4A:                 3 * PASCAL INITIALIZATION:
CA4A:                 4 ----------------------------------------
CA4A:              CA4A   5 PINIT1.0  EQU   *
CA4A:A9 22            6              LDA   #M.PASCAL+M.PAS1.0
CA4C:4C 51 CA         7              JMP   PINIT2
CA4F:              CA4F   8 PINIT     EQU   *
CA4F:A9 20            9              LDA   #M.PASCAL    ;SAY WE'RE
CA51:                10 *
CA51:              CA51  11 PINIT2    EQU   *
CA51:8D FB 04        12              STA   MODE         ;  RUNNING PASCAL
CA54:20 9B CD        13              JSR   FULL80       ;SET FULL 24x80 WINDOW
CA57:20 C8 CF        14              JSR   PSETUP       ;SETUP ZP STUFF
CA5A:                15 * BASE ADDR IS WRONG, BUT X.FF FIXES IT BELOW:
CA5A:                16 *  JSR BASCALC ;FORCE A GOOD BASCALC
CA5A:                17 *
CA5A:                18 * SEE IF THE CARD'S PLUGGED IN:
CA5A:                19 *
CA5A:20 24 CB        20              JSR   TESTCARD     ;IS IT THERE?
CA5D:F0 03   CA62    21              BEQ   PIGOOD       ;=>YES
CA5F:A2 09           22              LDX   #9           ;IORESULT='NO DEVICE'
CA61:60              23              RTS
CA62:                24 *
CA62:              CA62  25 PIGOOD    EQU   *
CA62:8D 01 C0        26              STA   SETB0COL     ;ENABLE 80 STORE
CA65:8D 0D C0        27              STA   SETB0VID     ;  AND 80 VIDEO
CA68:8D 0F C0        28              STA   SETALTCHAR   ;NORM+INV LCASE
CA6B:20 42 CD        29              JSR   X.FF         ;HOME & CLEAR IT
CA6E:20 DD CE        30              JSR   INVERT       ;PUT CURSOR THERE
CA71:A2 00           31              LDX   #0           ;IORESULT='GOOD'
CA73:60              32              RTS
CA74:                10              INCLUDE PREAD
CA74:                 2 ----------------------------------------
CA74:                 3 * PASCAL INPUT:
CA74:                 4 ----------------------------------------
CA74:              CA74   5 PREAD     EQU   *
CA74:20 C8 CF         6              JSR   PSETUP       ;SETUP ZP STUFF
CA77:                 7 *
CA77:20 15 CB         8              JSR   GETKEY       ;GET A KEYSTROKE
CA7A:29 7F            9              AND   #$7F         ;DROP HI BIT
CA7C:8D 7B 06        10              STA   CHAR         ;SAVE THE CHAR
CA7F:A2 00           11              LDX   #0           ;IORESULT='GOOD'
CA81:AD FB 04        12              LDA   MODE         ;ARE WE IN 1.0-MODE?
CA84:29 02           13              AND   #M.PAS1.0
CA86:F0 02   CA8A    14              BEQ   PREADRET2    ;=>NOPE
CA88:A2 C3           15              LDX   #<CN00       ;YES, RETURN CN IN X
CA8A:                16 *
CA8A:              CA8A  17 PREADRET2 EQU   *
CA8A:AD 7B 06        18              LDA   CHAR         ;RESTORE CHAR
CA8D:60              19              RTS
CA8E:                11              INCLUDE PWRITE
CA8E:                 2 ----------------------------------------
CA8E:                 3 * PASCAL OUTPUT:
CA8E:                 4 ----------------------------------------
CA8E:              CA8E   5 PWRITE    EQU   *
CA8E:8D 7B 06         6              STA   CHAR         ;SAVE CHARACTER
CA91:20 C8 CF         7              JSR   PSETUP       ;SETUP ZP STUFF
CA94:                 8 *
CA94:20 DD CE         9              JSR   INVERT       ;TURN CURSOR OFF
CA97:AD FB 04        10              LDA   MODE         ;ARE WE DOING GOTOXY?
CA9A:29 08           11              AND   #M.GOXY
CA9C:F0 2D   CACB    12              BEQ   PWRITE3      ;=>NO, PRINT IT
CA9E:                13 *
CA9E:                14 *  HANDLE GOTOXY STUFF:
CA9E:                15 *
CA9E:              CA9E  16 PWRITE2   EQU   *
CA9E:AD FB 06        17              LDA   XCOORD       ;ARE WE WAITING FOR X?
CAA1:10 0C   CAAF    18              BPL   GETY        ;=>NO, THIS IS Y
CAA3:AD 7B 06        19              LDA   CHAR
CAA6:38              20              SEC
CAA7:E9 20           21              SBC   #32          ;MAKE BINARY
CAA9:8D FB 06        22              STA   XCOORD
CAAC:4C 0F CB        23              JMP   PWRITERET    ;=>NOW WAIT FOR Y
CAAF:                24 *
```

```
CAAF:                    25 * NOW DO THE GOTOXY:
CAAF:                    26 *
CAAF:           CAAF     27 GETY         EQU   *
CAAF: AD 7B 06           28              LDA   CHAR          ;CONVERT YCOORD
CAB2: 38                 29              SEC
CAB3: E9 20              30              SBC   #32
CAB5: 8D FB 05           31              STA   OURCV
CAB8: 20 51 CB           32              JSR   BASCALC       ;COMPUTE BASE ADDRESS
CABB: AD FB 06           33              LDA   XCOORD
CABE: 8D 7B 05           34              STA   OURCH
CAC1: AD FB 04           35              LDA   MODE          ;TURN OFF GOTOXY
CAC4: 29 F7              36              AND   #255-M.GOXY
CAC6: 8D FB 04           37              STA   MODE
CAC9: D0 44    CB0F      38              BNE   PWRITERET     ;=>DONE (ALWAYS TAKEN)
CACB:                    39 *
CACB:           CACB     40 PWRITE3      EQU   *
CACB: AD 7B 06           41              LDA   CHAR          ;GET CHAR TO PRINT
CACE: C9 1E              42              CMP   #$1E          ;IS IT GOTOXY?
CAD0: F0 0A    CADC      43              BEQ   STARTXY       ;=>YES
CAD2: C9 20              44              CMP   #$20          ;IS IT OTHER CTL?
CAD4: B0 15    CAEB      45              BCS   PWRITE4       ;=>NO, PRINT IT
CAD6: 20 99 CB           46              JSR   CTLCHAR       ;EXECUTE IT IF POSSIBLE
CAD9: 4C 0F CB           47              JMP   PWRITERET     ;=>EXECUTED OR IGNORED
CADC:                    48 *
CADC:                    49 * START THE GOTOXY SEQUENCE:
CADC:                    50 *
CADC:           CADC     51 STARTXY      EQU   *
CADC: AD FB 04           52              LDA   MODE          ;TURN ON FLAG
CADF: 09 08              53              ORA   #M.GOXY
CAE1: 8D FB 04           54              STA   MODE
CAE4: A9 FF              55              LDA   #-1           ;SET X NEGATIVE TO
CAE6: 8D FB 06           56              STA   XCOORD        ; SHOW WE NEED IT
CAE9: 30 24    CB0F      57              BMI   PWRITERET     ;=>EXIT TILL COORDS COME BY (ALWAYS)
CAEB:                    58 *
CAEB:                    59 * JUST A PRINTABLE CHARACTER:
CAEB:                    60 *
CAEB:           CAEB     61 PWRITE4      EQU   *
CAEB: 09 80              62              ORA   #$80          ;FORCE TO NORMAL
CAED: AC 7B 05           63              LDY   OURCH         ;GET CH
CAF0: 20 F2 CE           64              JSR   STORCHAR      ;STUFF IT!
CAF3:                    65 *
CAF3:                    66 * BUMP CURSOR HORIZONTAL:
CAF3:                    67 *
CAF3: EE 7B 05           68              INC   OURCH         ;BUMP IT
CAF6: AD 7B 05           69              LDA   OURCH         ;ARE WE PAST THE
CAF9: C5 21              70              CMP   WNDWDTH       ; END OF THE LINE?
CAFB: 90 12    CB0F      71              BCC   PWRITERET     ;=>NO, NO PROBLEM
CAFD:                    72 *
CAFD:                    73 * IF IN TRANSPARENT MODE, DON'T
CAFD:                    74 *  WRAPAROUND THE RIGHT EDGE...
CAFD:                    75 *
CAFD: AD FB 04           76              LDA   MODE          ;GET MODE
CB00: 29 01              77              AND   #M.TRANS      ;WELL???
CB02: F0 05    CB09      78              BEQ   PWWRAP        ;=>NOT TRANSPARENT
CB04: CE 7B 05           79              DEC   OURCH         ;PIN AT RIGHT EDGE
CB07: D0 06    CB0F      80              BNE   PWRITERET     ;(ALWAYS TAKEN)
CB09:                    81 *
CB09:           CB09     82 PWWRAP       EQU   *
CB09: 20 EC CB           83              JSR   X.CR          ;YES, DO C/R
CB0C: 20 91 CC           84              JSR   X.LF          ; AND L/F
CB0F:                    85 *
CB0F:           CB0F     86 PWRITERET    EQU   *
CB0F: 20 DD CE           87              JSR   INVERT        ;TURN CURSOR ON
CB12: A2 00              88              LDX   #0            ;IORESULT='GOOD'
CB14: 60                 89              RTS
CB15:                    12              INCLUDE SUBS1
CB15:                     2 ----------------------------------------
CB15:                     3 * NAME     : GETKEY
CB15:                     4 * FUNCTION: GET A KEYSTROKE
CB15:                     5 * INPUT    : NONE
CB15:                     6 * OUTPUT   : AC=KEYCODE
CB15:                     7 * VOLATILE: NONE
CB15:                     8 ----------------------------------------
CB15:                     9 *
CB15:           CB15     10 GETKEY       EQU   *
CB15: E6 4E              11              INC   RNDL          ;BUMP RANDOM SEED
CB17: D0 02    CB1B      12              BNE   GETK2
CB19: E6 4F              13              INC   RNDH
CB1B:           CB1B     14 GETK2        EQU   *
CB1B: AD 00 C0           15              LDA   KBD           ;KEYPRESS?
CB1E: 10 F5    CB15      16              BPL   GETKEY        ;=>NOPE
CB20: 8D 10 C0           17              STA   KBDSTRB       ;CLEAR STROBE
CB23: 60                 18              RTS
CB24:                    19 ----------------------------------------
CB24:                    20 * NAME     : TESTCARD
CB24:                    21 * FUNCTION: SEE IF 80COL CARD PLUGGED IN
CB24:                    22 * INPUT    : NONE
CB24:                    23 * OUTPUT   : 'BEQ' IF CARD AVAILABLE
CB24:                    24 *          : 'BNE' IF NOT
CB24:                    25 * VOLATILE: AC,Y
CB24:                    26 ----------------------------------------
CB24:                    27 *
```

```
CB24:          CB24   28 TESTCARD    EQU   *
CB24: AD 1C C0        29              LDA   RDPAGE2      ;REMEMBER CURRENT VIDEO DISPLAY
CB27: 0A              30              ASL   A            ; IN THE CARRY
CB28: A9 88           31              LDA   #$88         ;USEFUL CHAR FOR TESTING
CB2A: 2C 1B C0        32              BIT   RD80COL      ;REMEMBER VIDEO MODE IN 'N'
CB2D: 8D 01 C0        33              STA   SET80COL     ;ENABLE 80COL STORE
CB30: 08              34              PHP                ;LOCK INTERRUPTS WHILE
CB31: 78              35              SEI                ; SCREENHOLES ARE WRONG
CB32: 08              36              PHP                ;SAVE 'N' AND 'C' FLAGS
CB33: 8D 55 C0        37              STA   TXTPAGE2     ;SET PAGE2
CB36: AC 00 04        38              LDY   $0400        ;GET FIRST CHAR
CB39: 8D 00 04        39              STA   $0400        ;SET TO A '*'
CB3C: AD 00 04        40              LDA   $0400        ;GET IT BACK FROM RAM
CB3F: 8C 00 04        41              STY   $0400        ;RESTORE ORIG CHAR
CB42: 28              42              PLP                ;RESTORE 'N' AND 'C' FLAGS
CB43: B0 03    CB48   43              BCS   STAY2        ;STAY IN PAGE2
CB45: 8D 54 C0        44              STA   TXTPAGE1     ;RESTORE PAGE1
CB48:          CB48   45 STAY2        EQU   *
CB48: 30 03    CB4D   46              BMI   STAY80       ;=>STAY IN 80COL MODE
CB4A: 8D 00 C0        47              STA   CLR80COL     ;TURN OFF 80COL STORE
CB4D:          CB4D   48 STAY80       EQU   *
CB4D: 28              49              PLP                ;ALLOW IRQ AGAIN
CB4E:          CB4E   50 TESTFAIL     EQU   *
CB4E: C9 88           51              CMP   #$88         ;WAS CHAR VALID?
CB50: 60              52              RTS                ;RETURN RESULT AS BEQ/BNE
CB51:                 53 ----------------------------------------
CB51:                 54 * NAME     : BASCALC,BASCALCZ
CB51:                 55 * FUNCTION: CALC BASE ADDR FOR SCREEN LINE
CB51:                 56 * INPUT    : OURCV (BASCALC)
CB51:                 57 *          : AC=CV (BASCALCZ)
CB51:                 58 * OUTPUT   : BASL/BASH
CB51:                 59 * VOLATILE: NOTHING
CB51:                 60 * CALLS    : SNIFFIRQ
CB51:                 61 ----------------------------------------
CB51:                 62 *
CB51:          FC75   63 SNIFFIRQ     EQU   $FC75
CB51:                 64 *
CB51:          CB51   65 BASCALC      EQU   *            ;RIPPED OFF FROM F8 ROM
CB51: 18              66              CLC                ;SHOW ENTRY POINT
CB52: 90 01    CB55   67              BCC   BSCLC1
CB54:          CB54   68 BASCALCZ     EQU   *
CB54: 38              69              SEC                ;SHOW ENTRY POINT
CB55:          CB55   70 BSCLC1       EQU   *
CB55: 48              71              PHA                ;SAVE AC
CB56: B0 03    CB5B   72              BCS   BSCLC1A      ;=>CV ALREADY IN AC
CB58: AD FB 05        73              LDA   OURCV
CB5B:          CB5B   74 BSCLC1A      EQU   *
CB5B: 48              75              PHA
CB5C: 4A              76              LSR   A
CB5D: 29 03           77              AND   #$03
CB5F: 09 04           78              ORA   #$04
CB61: 85 29           79              STA   BASH
CB63: 8D FB 07        80              STA   OLDBASH      ;SAVE FOR F/W PROTOCOL
CB66: 68              81              PLA
CB67: 29 18           82              AND   #$18
CB69: 90 02    CB6D   83              BCC   BSCLC2
CB6B: 69 7F           84              ADC   #$7F
CB6D: 85 28           85 BSCLC2      STA   BASL
CB6F: 0A              86              ASL   A
CB70: 0A              87              ASL   A
CB71: 05 28           88              ORA   BASL
CB73: 85 28           89              STA   BASL
CB75:                 90 *
CB75:                 91 * HANDLE THE SCROLLING WINDOW:
CB75:                 92 *
CB75: A5 20           93              LDA   WNDLFT
CB77: 08              94              PHP                ;PRESERVE CARRY
CB78: 2C 1F C0        95              BIT   RD80VID      ;WHICH MODE?
CB7B: 10 01    CB7E   96              BPL   BASCLC3      ;=>40.  NO DIVIDE.
CB7D: 4A              97              LSR   A            ;DIVIDE BY 2 FOR 80COL WINDOW
CB7E:          CB7E   98 BASCLC3      EQU   *
CB7E: 28              99              PLP                ;RESTORE CARRY
CB7F: 65 28          100              ADC   BASL         ;ADJUST BASE FOR WNDLFT
CB81: 85 28          101              STA   BASL
CB83: 8D 7B 07       102              STA   OLDBASL      ;SAVE FOR F/W PROTOCOL
CB86:                103 *
CB86:                104 * SNIFF FOR IRQ IF NECESSARY:
CB86:                105 *
CB86: AD FB 04       106              LDA   MODE
CB89: 29 01          107              AND   #M.IRQ
CB8B: F0 0A    CB97  108              BEQ   BASCLCX      ;=>IRQ DISABLED, RETURN
CB8D: AD FB 04       109              LDA   MODE         ;IS BASIC RUNNING?
CB90: 29 20          110              AND   #M.PASCAL
CB92: D0 03    CB97  111              BNE   BASCLCX      ;=>DON'T SNIFF UNDER PASCAL
CB94: 20 75 FC       112              JSR   SNIFFIRQ     ;GO DO IT
CB97:          CB97  113 BASCLCX      EQU   *
CB97: 68             114              PLA                ;RESTORE AC
CB98: 60             115              RTS
CB99:                116 ----------------------------------------
CB99:                117 * NAME     : CTLCHAR
CB99:                118 * FUNCTION: EXECUTE CTL CHAR
CB99:                119 * INPUT    : AC=CHAR
```

```
CB99:                  120 * OUTPUT   : 'BCS' IF NOT CTL
CB99:                  121 *          : 'BCC' IF CTL EXECUTED
CB99:                  122 * VOLATILE: NOTHING
CB99:                  123 * CALLS    : MANY THINGS
CB99:                  124 ----------------------------------------
CB99:                  125 *
CB99:          CB99    126 CTLCHAR     EQU   *
CB99: 8D 7B 04         127             STA   TEMP1          ;TEMP SAVE OF CHAR
CB9C: 48               128             PHA                  ;SAVE AC
CB9D: 98               129             TYA                  ;SAVE Y
CB9E: 48               130             PHA
CB9F:                  131 *
CB9F: AC 7B 04         132             LDY   TEMP1          ;GET CHAR IN QUESTION
CBA2: C0 07            133             CPY   #$07           ;IS IT NUL..ACK?
CBA4: 90 05    CBAB    134             BCC   CTLCHARX       ;=>YES, NOT USED
CBA6: B9 71 CC         135             LDA   CTLADH-7,Y     ;IS IT CTL?
CBA9: D0 03    CBAE    136             BNE   CTLGO          ;=>YES
CBAB:          CBAB    137 CTLCHARX    EQU   *
CBAB: 38               138             SEC                  ;SAY 'NOT CTL'
CBAC: B0 04    CBB2    139             BCS   CTLRET         ;=>DONE
CBAE:                  140 *
CBAE:          CBAE    141 CTLGO       EQU   *
CBAE: 20 B6 CB         142             JSR   CTLXFER        ;EXECUTE SUBROUTINE
CBB1:                  143 *
CBB1: 18               144             CLC                  ;SAY 'CTL CHAR EXECUTED'
CBB2:          CBB2    145 CTLRET      EQU   *
CBB2: 68               146             PLA                  ;RESTORE
CBB3: A8               147             TAY                  ; Y
CBB4: 68               148             PLA                  ;  AND AC
CBB5: 60               149             RTS
CBB6:                  150 *
CBB6:          CBB6    151 CTLXFER     EQU   *
CBB6: 48               152             PHA                  ;PUSH ONTO STACK FOR
CBB7: B9 5B CC         153             LDA   CTLADL-7,Y     ; TRANSFER TRICK
CBBA: 48               154             PHA
CBBB: 60               155             RTS                  ;XFER TO ROUTINE
CBBC:                  156 *
CBBC:                  157 * EXECUTE BELL:
CBBC:                  158 *
CBBC:          CBBC    159 X.BELL      EQU   *
CBBC: A9 40            160             LDA   #$40           ;RIPPED OFF FROM MONITOR
CBBE: 20 CF CB         161             JSR   WAIT
CBC1: A0 C0            162             LDY   #$C0
CBC3: A9 0C            163 BELL2       LDA   #$0C
CBC5: 20 CF CB         164             JSR   WAIT
CBC8: AD 30 C0         165             LDA   SPKR
CBCB: 88               166             DEY
CBCC: D0 F5    CBC3    167             BNE   BELL2
CBCE: 60               168             RTS
CBCF:                  169 *
CBCF:          CBCF    170 WAIT        EQU   *              ;RIPPED OFF FROM MONITOR ROM
CBCF: 38               171             SEC
CBD0: 48               172 WAIT2       PHA
CBD1: E9 01            173 WAIT3       SBC   #1
CBD3: D0 FC    CBD1    174             BNE   WAIT3
CBD5: 68               175             PLA
CBD6: E9 01            176             SBC   #1
CBD8: D0 F6    CBD0    177             BNE   WAIT2
CBDA: 60               178             RTS
CBDB:                  179 *
CBDB:                  180 * EXECUTE BACKSPACE:
CBDB:                  181 *
CBDB:          CBDB    182 X.BS        EQU   *
CBDB: CE 7B 05         183             DEC   OURCH          ;BACK UP CH
CBDE: 10 0B    CBEB    184             BPL   BSDONE         ;=>DONE
CBE0: A5 21            185             LDA   WNDWDTH        ;BACK UP TO PRIOR LINE
CBE2:          CBE2    186 BS40        EQU   *
CBE2: 8D 7B 05         187             STA   OURCH          ;SET CH
CBE5: CE 7B 05         188             DEC   OURCH
CBE8: 20 34 CC         189             JSR   X.VS           ;NOW DO REV LINEFEED
CBEB:          CBEB    190 BSDONE      EQU   *
CBEB: 60               191             RTS
CBEC:                  192 *
CBEC:                  193 * EXECUTE CARRIAGE RETURN:
CBEC:                  194 *
CBEC:          CBEC    195 X.CR        EQU   *
CBEC: AD FB 04         196             LDA   MODE           ;WHICH LANGUAGE?
CBEF: 29 20            197             AND   #M.PASCAL
CBF1: D0 0A    CBFD    198             BNE   X.CRPAS        ;=>PASCAL, NO CLR EOL
CBF3: AD FB 04         199             LDA   MODE           ;INPUT OR OUTPUT?
CBF6: 29 40            200             AND   #M.BINPUT
CBF8: F0 03    CBFD    201             BEQ   X.CRPAS        ;=>OUTPUT. NO CLEARING
CBFA: 20 48 CD         202             JSR   X.GS           ;CLEAR TO EOL
CBFD:                  203 *
CBFD:          CBFD    204 X.CRPAS     EQU   *
CBFD: A9 00            205             LDA   #0             ;BACK UP CH TO
CBFF: 8D 7B 05         206             STA   OURCH          ; BEGINNING OF LINE
CC02: AD FB 04         207             LDA   MODE           ;ARE WE IN BASIC?
CC05: 29 20            208             AND   #M.PASCAL
CC07: D0 03    CC0C    209             BNE   X.CRRET        ;=>PASCAL, AVOID AUTO L/F
CC09: 20 91 CC         210             JSR   X.LF           ;EXECUTE AUTO LF FOR BASIC
CC0C:          CC0C    211 X.CRRET     EQU   *
```

```
CC0C:60            212              RTS
CC0D:       0000   213              DO    0           ;NO MORE ROM SPACE!
 S                 214 *
 S                 215 *            EXECUTE SYNC:
 S                 216 *
 S                 217 X.SYN        EQU   *
 S                 218              LDA   RDVBLBAR    ;WAIT FOR VBL
 S                 219              BPL   X.SYN       ;=>WAIT FOR VIDEO SCAN
 S                 220 X.SYN2       LDA   RDVBLBAR    ;NOW WAIT FOR
 S                 221              BMI   X.SYN2      ; BLANKING TO BEGIN
 S                 222              RTS
CC0D:              223              FIN
CC0D:              224 *
CC0D:              225 * EXECUTE HOME:
CC0D:              226 *
CC0D:       CC0D   227 X.EM         EQU   *
CC0D:A5 22         228              LDA   WNDTOP
CC0F:8D FB 05      229              STA   OURCV       ;STUFF CV
CC12:A9 00         230              LDA   #0
CC14:8D 7B 05      231              STA   OURCH       ;STUFF CH
CC17:4C 51 CB      232              JMP   BASCALC     ;RETURN VIA BASCALC (UGH!)
CC1A:              233 *
CC1A:              234 * EXECUTE CLEAR LINE:
CC1A:              235 *
CC1A:       CC1A   236 X.SUB        EQU   *
CC1A:A4 21         237              LDY   WNDWDTH
CC1C:88            238              DEY
CC1D:       CC1D   239 X.SUB80      EQU   *
CC1D:A9 A0         240              LDA   #' '        ;BLANKIE BLANK
CC1F:       CC1F   241 X.SUBLP      EQU   *
CC1F:20 F2 CE      242              JSR   STORCHAR    ;STUFF THE BLANK
CC22:88            243              DEY
CC23:10 FA   CC1F  244              BPL   X.SUBLP     ;=>CLEAR THE LINE
CC25:60            245              RTS
CC26:              246 *
CC26:              247 * EXECUTE FORWARD SPACE:
CC26:              248 *
CC26:       CC26   249 X.FS         EQU   *
CC26:EE 7B 05      250              INC   OURCH       ;BUMP CH
CC29:AD 7B 05      251              LDA   OURCH       ;GET THE POSITION
CC2C:C5 21         252              CMP   WNDWDTH     ;OFF THE RIGHT SIDE?
CC2E:90 03   CC33  253              BCC   X.FSRET     ;=>NO, GOOD
CC30:20 EC CB      254              JSR   X.CR        ;=>YES, WRAP AROUND
CC33:              255 *
CC33:       CC33   256 X.FSRET      EQU   *
CC33:60            257              RTS
CC34:              258 *
CC34:              259 * EXECUTE REVERSE LINEFEED:
CC34:              260 *
CC34:       CC34   261 X.US         EQU   *
CC34:CE FB 05      262              DEC   OURCV       ;BACK UP CV
CC37:30 07   CC40  263              BMI   X.US1       ;=>OFF TOP OF SCREEN
CC39:AD FB 05      264              LDA   OURCV
CC3C:C5 22         265              CMP   WNDTOP      ;OFF TOP OF WINDOW?
CC3E:B0 05   CC45  266              BCS   X.US2       ;=>NO, STILL IN WINDOW
CC40:              267 *
CC40:              268 * PIN CV TO WINDOW TOP:
CC40:              269 *
CC40:       CC40   270 X.US1        EQU   *
CC40:EE FB 05      271              INC   OURCV       ;PUT BACK WHERE IT WAS
CC43:F0 03   CC48  272              BEQ   X.USRET     ;IT GOES TO 0 ALWAYS
CC45:       CC45   273 X.US2        EQU   *
CC45:20 51 CB      274              JSR   BASCALC     ;RECOMPUTE BASE ADDR
CC48:       CC48   275 X.USRET      EQU   *
CC48:60            276              RTS
CC49:              277 *
CC49:              278 * EXECUTE "NORMAL VIDEO"
CC49:              279 *
CC49:       CC49   280 X.SO         EQU   *
CC49:AD FB 04      281              LDA   MODE        ;SET MODE BIT
CC4C:29 FB         282              AND   #255-M.VMODE ;SET 'NORMAL'
CC4E:A0 FF         283              LDY   #255
CC50:D0 07   CC59  284              BNE   STUFFINV    ;(ALWAYS)
CC52:              285 *
CC52:              286 * EXECUTE "INVERSE VIDEO"
CC52:              287 *
CC52:       CC52   288 X.SI         EQU   *
CC52:AD FB 04      289              LDA   MODE        ;SET MODE BIT
CC55:09 04         290              ORA   #M.VMODE    ;SET 'INVERSE'
CC57:A0 7F         291              LDY   #127
CC59:       CC59   292 STUFFINV     EQU   *
CC59:8D FB 04      293              STA   MODE        ;SET MODE
CC5C:84 32         294              STY   INVFLG      ;STUFF FLAG TOO
CC5E:60            295              RTS
CC5F:       CC5F   297 CTLADL       EQU   *
CC5F:BB            298              DFB   >X.BELL-1   ;BEL
CC60:DA            299              DFB   >X.BS-1     ;BS
CC61:00            300              DFB   0           ;HT
CC62:90            301              DFB   >X.LF-1     ;LF
CC63:22            302              DFB   >X.VT-1     ;VT
CC64:41            303              DFB   >X.FF-1     ;FF
CC65:EB            304              DFB   >X.CR-1     ;CR
```

```
CC66:48             305              DFB   >X.SO-1       ;SO
CC67:51             306              DFB   >X.SI-1       ;SI
CC68:00             307              DFB   0             ;DLE
CC69:58             308              DFB   >X.DC1-1      ;DC1
CC6A:76             309              DFB   >X.DC2-1      ;DC2
CC6B:00             310              DFB   0             ;DC3
CC6C:00             311              DFB   0             ;DC4
CC6D:BF             312              DFB   >X.NAK-1      ;NAK
CC6E:A9             313              DFB   >SCROLLDN-1   ;SYN
CC6F:A3             314              DFB   >SCROLLUP-1   ;ETB
CC70:00             315              DFB   0             ;CAN
CC71:0C             316              DFB   >X.EM-1       ;EM
CC72:19             317              DFB   >X.SUB-1      ;SUB
CC73:00             318              DFB   0             ;ESC
CC74:25             319              DFB   >X.FS-1       ;FS
CC75:47             320              DFB   >X.GS-1       ;GS
CC76:00             321              DFB   0             ;RS
CC77:33             322              DFB   >X.US-1       ;US
CC78:               323 *
CC78:        CC78   324 CTLADH       EQU   *
CC78:CB             325              DFB   <X.BELL-1     ;BEL
CC79:CB             326              DFB   <X.BS-1       ;BS
CC7A:00             327              DFB   0             ;HT
CC7B:CC             328              DFB   <X.LF-1       ;LF
CC7C:CD             329              DFB   <X.VT-1       ;VT
CC7D:CD             330              DFB   <X.FF-1       ;FF
CC7E:CB             331              DFB   <X.CR-1       ;CR
CC7F:CC             332              DFB   <X.SO-1       ;SO
CC80:CC             333              DFB   <X.SI-1       ;SI
CC81:00             334              DFB   0             ;DLE
CC82:CD             335              DFB   <X.DC1-1      ;DC1
CC83:CD             336              DFB   <X.DC2-1      ;DC2
CC84:00             337              DFB   0             ;DC3
CC85:00             338              DFB   0             ;DC4
CC86:CD             339              DFB   <X.NAK-1      ;NAK
CC87:CC             340              DFB   <SCROLLDN-1   ;SYN
CC88:CC             341              DFB   <SCROLLUP-1   ;ETB
CC89:00             342              DFB   0             ;CAN
CC8A:CC             343              DFB   <X.EM-1       ;EM
CC8B:CC             344              DFB   <X.SUB-1      ;SUB
CC8C:00             345              DFB   0             ;ESC
CC8D:CC             346              DFB   <X.FS-1       ;FS
CC8E:CD             347              DFB   <X.GS-1       ;GS
CC8F:00             348              DFB   0             ;RS
CC90:CC             349              DFB   <X.US-1       ;US
CC91:                13              INCLUDE SUBS2
CC91:                 2 *
CC91:                 3 * EXECUTE LINEFEED:
CC91:                 4 *
CC91:        CC91     5 X.LF          EQU   *
CC91:EE FB 05         6               INC   OURCV         ;BUMP CV
CC94:AD FB 05         7               LDA   OURCV         ;SEE IF OFF BOTTOM
CC97:C5 23            8               CMP   WNDBTM        ;OFF THE END?
CC99:B0 03   CC9E     9               BCS   X.LF2         ;=>YES
CC9B:4C 20 CD        10               JMP   X.LFRET       ;=>NO, DONE
CC9E:        CC9E    11 X.LF2         EQU   *
CC9E:A4 23           12               LDY   WNDBTM        ;SET TO
CCA0:88              13               DEY
CCA1:8C FB 05        14               STY   OURCV         ; THE BOTTOM
CCA4:                15 *
CCA4:                16 * SCROLL THE SCREEN:
CCA4:                17 *
CCA4:        CCA4    18 SCROLLUP      EQU   *
CCA4:8A              19               TXA                 ;SAVE X
CCA5:48              20               PHA
CCA6:A2 01           21               LDX   #1            ;DIRECTION=UP
CCA8:D0 04   CCAE    22               BNE   SCROLL1
CCAA:        CCAA    23 SCROLLDN      EQU   *
CCAA:8A              24               TXA                 ;SAVE X
CCAB:48              25               PHA
CCAC:A2 00           26               LDX   #0            ;DIRECTION=DOWN
CCAE:                27 *
CCAE:        CCAE    28 SCROLL1       EQU   *
CCAE:2C 1F C0        29               BIT   RDBOVID       ;WHICH MODE?
CCB1:10 05   CCB8    30               BPL   SCROLL2       ;=>40.  DO WITH EXISTING WIDTH
CCB3:A5 21           31               LDA   WNDWDTH       ;TEMPORARILY SAVE
CCB5:48              32               PHA                 ; THE WIDTH AND
CCB6:46 21           33               LSR   WNDWDTH       ;  DIVIDE IT BY 2
CCB8:                34 *
CCB8:        CCB8    35 SCROLL2       EQU   *
CCB8:20 D1 CC        36               JSR   SCRLSUB       ;SCROLL 40 COLS
CCBB:2C 1F C0        37               BIT   RDBOVID       ;ARE WE IN 80-MODE?
CCBE:10 51   CD11    38               BPL   X.SCRLRET     ;=>NO, DONE
CCC0:                39 *
CCC0:                40 * FOR 80, DO THE OTHER PAGE...
CCC0:                41 *
CCC0:08              42               PHP                 ;ENSURE IRQ INHIBITED
CCC1:78              43               SEI                 ; WHILE TXTPAGE2 MAPPED IN
CCC2:AD 55 C0        44               LDA   TXTPAGE2      ;SET PAGE2
CCC5:20 D1 CC        45               JSR   SCRLSUB       ;SCROLL PAGE 2
CCC8:AD 54 C0        46               LDA   TXTPAGE1      ;RESTORE PAGE1
CCCB:28              47               PLP                 ;RESTORE IRQ STATE NOW
```

```
CCCC:68             48              PLA
CCCD:85 21          49              STA   WNDWDTH
CCCF:D0 40    CD11  50              BNE   X.SCRLRET   ;=>DONE SCROLL80 (ALWAYS TAKEN)
CCD1:               51 *
CCD1:               52 * 40-COLUMN WINDOWED SCROLL:
CCD1:               53 *
CCD1:         CCD1  54 SCRLSUB      EQU   *
CCD1:BC F9 CF       55              LDY   WNDTAB,X    ;GET WINDOW TOP/BOT
CCD4:B9 00 00       56              LDA   0,Y
CCD7:E0 01          57              CPX   #1          ;SCROLLING UP?
CCD9:B0 02    CCDD  58              BCS   MSCRL0      ;=>YES, NO PROBLEM
CCDB:E9 00          59              SBC   #0          ;-1 IF DOWN (SRC=BTM-1)
CCDD:         CCDD  60 MSCRL0       EQU   *
CCDD:48             61              PHA
CCDE:20 54 CB       62              JSR   BASCALCZ
CCE1:A5 28          63 MSCRL1       LDA   BASL
CCE3:85 2A          64              STA   BAS2L
CCE5:A5 29          65              LDA   BASH
CCE7:85 2B          66              STA   BAS2H
CCE9:A4 21          67              LDY   WNDWDTH
CCEB:88             68              DEY
CCEC:68             69              PLA
CCED:1B             70              CLC
CCEE:7D F0 CF       71              ADC   PLUSMINUS1,X ;UP/DOWN
CCF1:D5 22          72              CMP   WNDTOP,X    ;AT THE END?
CCF3:F0 0D    CD02  73              BEQ   MSCRLRET
CCF5:48             74              PHA
CCF6:20 54 CB       75              JSR   BASCALCZ
CCF9:B1 28          76 MSCRL2       LDA   (BASL),Y
CCFB:91 2A          77              STA   (BAS2L),Y
CCFD:88             78              DEY
CCFE:10 F9    CCF9  79              BPL   MSCRL2
CD00:30 DF    CCE1  80              BMI   MSCRL1
CD02:               81 *
CD02:         CD02  82 MSCRLRET     EQU   *
CD02:E0 00          83              CPX   #0          ;SCROLLING DOWN?
CD04:D0 0A    CD10  84              BNE   MSCRLRTS    ;=>NO
CD06:20 54 CB       85              JSR   BASCALCZ
CD09:         CD09  86 ONEMORE      EQU   *
CD09:B1 28          87              LDA   (BASL),Y
CD0B:91 2A          88              STA   (BAS2L),Y
CD0D:88             89              DEY
CD0E:10 F9    CD09  90              BPL   ONEMORE
CD10:         CD10  91 MSCRLRTS     EQU   *
CD10:60             92              RTS
CD11:               93 *
CD11:               94 * DONE WITH THE SCROLLING JAZZ:
CD11:               95 *
CD11:         CD11  96 X.SCRLRET    EQU   *
CD11:B4 22          97              LDY   WNDTOP,X    ;CLEAR TOP OR BOTTOM LINE
CD13:8A             98              TXA               ;IF GETTING TOP,
CD14:F0 01    CD17  99              BEQ   X.SCRLRET2  ; DON'T DECREMENT!
CD16:88            100              DEY
CD17:         CD17 101 X.SCRLRET2   EQU   *
CD17:98            102              TYA               ;TEMP CV SETUP
CD18:20 54 CB      103              JSR   BASCALCZ    ;COMPUTE BASE OF LINE TO CLEAR
CD1B:68            104              PLA               ;RESTORE
CD1C:AA            105              TAX               ; X
CD1D:20 1A CC      106              JSR   X.SUB       ;CLEAR BOTTOM LINE
CD20:              107 *
CD20:         CD20 108 X.LFRET      EQU   *
CD20:4C 51 CB      109              JMP   BASCALC     ;RETURN VIA BASCALC (UGH!)
CD23:              110 *
CD23:              111 * EXECUTE CLR TO EOS:
CD23:              112 *
CD23:         CD23 113 X.VT         EQU   *
CD23:20 48 CD      114              JSR   X.GS        ;CLEAR TO EOL
CD26:AD FB 05      115              LDA   OURCV       ;SAVE CV
CD29:48            116              PHA
CD2A:10 06    CD32 117              BPL   X.VTNEXT    ;DO NEXT LINE (ALWAYS TAKEN)
CD2C:         CD2C 118 X.VTLOOP     EQU   *
CD2C:20 51 CB      119              JSR   BASCALC     ;BASCALC IT
CD2F:20 1A CC      120              JSR   X.SUB       ;CLEAR LINE
CD32:         CD32 121 X.VTNEXT     EQU   *
CD32:EE FB 05      122              INC   OURCV       ;BUMP CV
CD35:AD FB 05      123              LDA   OURCV
CD38:C5 23         124              CMP   WNDBTM      ;OFF SCREEN?
CD3A:90 F0    CD2C 125              BCC   X.VTLOOP    ;=>NO, KEEP GOING
CD3C:68            126              PLA               ;RESTORE
CD3D:8D FB 05      127              STA   OURCV       ; CV
CD40:10 DE    CD20 128              BPL   X.LFRET     ;RETURN VIA SIMILAR CODE
CD42:              129 *
CD42:              130 * EXECUTE CLEAR:
CD42:              131 *
CD42:         CD42 132 X.FF         EQU   *
CD42:20 0D CC      133              JSR   X.EM        ;HOME THE CURSOR
CD45:4C 23 CD      134              JMP   X.VT        ;RETURN VIA CLREOS (UGH!)
CD48:              135 *
CD48:              136 * EXECUTE CLEAR TO EOL:
CD48:              137 *
CD48:         CD48 138 X.GS         EQU   *
```

```
CD4B:AC 7B 05       139              LDY   OURCH        ;GET CH
CD4B:4C 54 CD       140              JMP   X.GS2        ;CHECK FOR END FIRST!
CD4E:         CD4E  141 X.GSEOLZ     EQU   *            ;FER U HACKERS
CD4E:A9 A0          142              LDA   #'           '
CD50:20 F2 CE       143              JSR   STORCHAR     ;STUFF IT
CD53:C8             144              INY
CD54:         CD54  145 X.GS2        EQU   *
CD54:C4 21          146              CPY   WNDWDTH      ;STOP SOMETIME
CD56:90 F6    CD4E  147              BCC   X.GSEOLZ     ;YASL DO MORE
CD58:60             148              RTS
CD59:               149 *
CD59:               150 * EXECUTE '40COL MODE':
CD59:               151 *
CD59:         CD59  152 X.DC1        EQU   *
CD59:A9 00          153              LDA   #0           ;ASSUME TEXTMODE
CD5B:85 20          154              STA   WNDLFT
CD5D:2C 1A C0       155              BIT   RDTEXT       ;ARE WE IN TEXT MODE?
CD60:30 02    CD64  156              BMI   X.DC1B       ;=>YES
CD62:A9 14          157              LDA   #20          ;IF GR, SET SPLITSCREEN
CD64:         CD64  158 X.DC1B       EQU   *
CD64:85 22          159              STA   WNDTOP
CD66:A9 18          160              LDA   #24
CD68:85 23          161              STA   WNDBTM
CD6A:A9 28          162              LDA   #40
CD6C:85 21          163              STA   WNDWDTH
CD6E:2C 1F C0       164              BIT   RD80VID      ;WERE WE IN 80-MODE?
CD71:10 03    CD76  165              BPL   X.DC1RTS     ;=>NO, NO CVT NEEDED
CD73:20 DB CD       166              JSR   SCRN84       ;CVT 80-->40
CD76:         CD76  167 X.DC1RTS     EQU   *
CD76:60             168              RTS
CD77:               169 *
CD77:               170 * EXECUTE '80COL MODE':
CD77:               171 *
CD77:         CD77  172 X.DC2        EQU   *
CD77:20 24 CB       173              JSR   TESTCARD     ;IS CARD THERE?
CD7A:D0 1E    CD9A  174              BNE   X.DC2RET     ;=>NOPE, FORGET IT
CD7C:20 9B CD       175              JSR   FULL80       ;SET FULL WINDOW
CD7F:2C 1A C0       176              BIT   RDTEXT       ;ARE WE IN TEXT MODE?
CD82:30 04    CD88  177              BMI   X.DC2B       ;=>YES
CD84:A9 14          178              LDA   #20          ;IF GR, SET SPLITSCREEN
CD86:85 22          179              STA   WNDTOP
CD88:         CD88  180 X.DC2B       EQU   *
CD88:2C 1B C0       181              BIT   RD80COL      ;REMEMBER PRIOR MODE
CD8B:30 0D    CD9A  182              BMI   X.DC2RET     ;=>NO CVT NEEDED IF WAS 80
CD8D:4C 32 CE       183              JMP   SCRN48       ;RET VIA CONVERT 40-->80
CD90:               184 *
CD90:               185 * EXECUTE 'QUIT':
CD90:               186 *
CD90:         CD90  187 X.NAK        EQU   *
CD90:AD FB 04       188              LDA   MODE         ;ONLY VALID IN BASIC
CD93:29 20          189              AND   #M.PASCAL
CD95:D0 03    CD9A  190              BNE   X.NAKRET     ;IGNORE IF PASCAL
CD97:20 AA CD       191              JSR   QUIT         ;GET SETUP TO QUIT
CD9A:         CD9A  192 X.NAKRET     EQU   *
CD9A:         CD9A  193 X.DC2RET     EQU   *
CD9A:60             194              RTS                ;DONE; CALLER WON'T RETURN
CD9B:               195 ----------------------------------------
CD9B:               196 * NAME     : FULL80
CD9B:               197 * FUNCTION: SET FULL 80COL WINDOW
CD9B:               198 * INPUT    : NONE
CD9B:               199 * OUTPUT   : WINDOW PARAMETERS
CD9B:               200 * VOLATILE: AC
CD9B:               201 ----------------------------------------
CD9B:               202 *
CD9B:         CD9B  203 FULL80       EQU   *
CD9B:A9 00          204              LDA   #0
CD9D:85 22          205              STA   WNDTOP
CD9F:85 20          206              STA   WNDLFT
CDA1:A9 50          207              LDA   #80
CDA3:85 21          208              STA   WNDWDTH
CDA5:A9 18          209              LDA   #24
CDA7:85 23          210              STA   WNDBTM
CDA9:60             211              RTS
CDAA:               212 ----------------------------------------
CDAA:               213 * NAME     : QUIT
CDAA:               214 * FUNCTION: SETUP TO QUIT THE CARD
CDAA:               215 * INPUT    : NOTHING
CDAA:               216 * OUTPUT   : NOTHING
CDAA:               217 * VOLATILE: ALL REGS
CDAA:               218 * CALLS    : X.FF,FULL80,BASCALC
CDAA:               219 *          : SETKBD,SETVID
CDAA:               220 ----------------------------------------
CDAA:               221 *
CDAA:         CDAA  222 QUIT         EQU   *
CDAA:A9 00          223              LDA   #0           ;SET FULL 40-COL WINDOW
CDAC:85 22          224              STA   WNDTOP
CDAE:85 20          225              STA   WNDLFT
CDB0:A9 18          226              LDA   #24
CDB2:85 23          227              STA   WNDBTM
CDB4:A9 28          228              LDA   #40
CDB6:85 21          229              STA   WNDWDTH
```

```
CDB8:2C 1F C0      230              BIT   RDB0VID      ;WHAT WIDTH?
CDBB:10 03   CDC0  231              BPL   QUIT2        ;=>NO CVT NEEDED IF 40
CDBD:20 DB CD      232              JSR   SCRN84       ;CONVERT 40-->80
CDC0:              233 *
CDC0:        CDC0  234 QUIT2        EQU   *
CDC0:A9 17         235              LDA   #23          ;VTAB TO THE
CDC2:8D FB 05      236              STA   OURCV        ; BOTTOM LINE
CDC5:20 51 CB      237              JSR   BASCALC
CDC8:A9 00         238              LDA   #0           ;   AND PLACE CURSOR
CDCA:8D 7B 05      239              STA   OURCH        ;    AT LEFT SIDE
CDCD:8D 0E C0      240              STA   CLRALTCHAR   ;LCASE CHARS OFF
CDD0:A9 FF         241              LDA   #$FF         ;DESTROY THE
CDD2:8D FB 04      242              STA   MODE         ; MODE BYTE
CDD5:20 93 FE      243              JSR   SETVID       ;PR#0
CDD8:4C B9 FE      244              JMP   SETKBD       ;RETURN VIA IN#0 (UGH!)
CDDB:              245 ---------------------------------------
CDDB:              246 * NAME     : SCRN84
CDDB:              247 * FUNCTION: CONVERT 80VID-->40VID
CDDB:              248 * INPUT    : NONE
CDDB:              249 * OUTPUT   : NONE
CDDB:              250 * VOLATILE: ALL REGISTERS
CDDB:              251 * NOTE     : USES 'BAS2H/L' AS TEMPS
CDDB:              252 ---------------------------------------
CDDB:              253 *
CDDB:        CDDB  254 SCRN84       EQU   *
CDDB:AD FB 05      255              LDA   OURCV        ;SAVE CURRENT
CDDE:48            256              PHA
CDDF:AD 7B 05      257              LDA   OURCH        ; SETTINGS
CDE2:48            258              PHA
CDE3:              259 *
CDE3:A9 17         260              LDA   #23
CDE5:85 2A         261              STA   BAS2L        ;USE AS A TEMP
CDE7:8D 01 C0      262              STA   SET80COL
CDEA:A5 2A         263 SCR40        LDA   BAS2L
CDEC:20 54 CB      264              JSR   BASCALCZ     ;BEGIN AT BOTTOM AND WORK UP
CDEF:20 0A CE      265              JSR   ATEFOR       ;DO THIS LINE
CDF2:C6 2A         266              DEC   BAS2L
CDF4:30 0B   CE01  267              BMI   SCR40RET     ;=>DONE (HIT TOP)
CDF6:2C 1A C0      268              BIT   RDTEXT       ;ARE WE IN MIDEXMODE?
CDF9:30 EF   CDEA  269              BMI   SCR40        ;=>NO, DO ENTIRE SCREEN
CDFB:A5 2A         270              LDA   BAS2L        ;IF SO, ONLY DO BOTTOM
CDFD:C9 14         271              CMP   #20          ; FOUR (4) LINES OF WINDOW
CDFF:B0 E9   CDEA  272              BCS   SCR40
CE01:        CE01  273 SCR40RET     EQU   *
CE01:8D 00 C0      274              STA   CLR80COL
CE04:8D 0C C0      275              STA   CLR80VID
CE07:4C 58 CE      276              JMP   SCRNRET      ;RETURN VIA SIMILAR CODE
CE0A:              277 *
CE0A:        CE0A  278 ATEFOR       EQU   *
CE0A:08            279              PHP                ;LOCK IRQ WHILE
CE0B:78            280              SEI                ; SCREENHOLES ARE WRONG
CE0C:A0 28         281              LDY   #40
CE0E:84 2B         282              STY   BAS2H
CE10:2C 54 C0      283              BIT   TXTPAGE1
CE13:20 22 CE      284 ATEFOR1      JSR   GET84
CE16:2C 55 C0      285              BIT   TXTPAGE2
CE19:20 22 CE      286              JSR   GET84
CE1C:A4 2B         287              LDY   BAS2H        ;DONE?
CE1E:D0 F3   CE13  288              BNE   ATEFOR1      ;=>NO, DO WHOLE LINE
CE20:28            289              PLP                ;RESTORE IRQ NOW
CE21:60            290              RTS
CE22:              291 *
CE22:C6 2B         292 GET84        DEC   BAS2H
CE24:A5 2B         293              LDA   BAS2H
CE26:4A            294              LSR   A
CE27:A8            295              TAY
CE28:B1 28         296              LDA   (BASL),Y
CE2A:A4 2B         297              LDY   BAS2H
CE2C:2C 54 C0      298              BIT   TXTPAGE1
CE2F:91 28         299              STA   (BASL),Y
CE31:60            300              RTS
CE32:              301 ---------------------------------------
CE32:              302 * NAME     : SCRN48
CE32:              303 * FUNCTION: CONVERT 40VID-->80VID
CE32:              304 * INPUT    : NONE
CE32:              305 * OUTPUT   : NONE
CE32:              306 * VOLATILE: ALL REGISTERS
CE32:              307 * NOTE     : USES 'BAS2H/L' AS TEMPS
CE32:              308 ---------------------------------------
CE32:              309 *
CE32:        CE32  310 SCRN48       EQU   *
CE32:AD FB 05      311              LDA   OURCV        ;SAVE CV
CE35:48            312              PHA
CE36:AD 7B 05      313              LDA   OURCH        ; AND CH
CE39:48            314              PHA
CE3A:              315 *
CE3A:A9 17         316              LDA   #23
CE3C:85 2A         317              STA   BAS2L        ;USE AS A TEMP
CE3E:A5 2A         318 SCR80        LDA   BAS2L
CE40:20 54 CB      319              JSR   BASCALCZ     ;BEGIN AT BOTTOM AND WORK UP
CE43:20 63 CE      320              JSR   FORATE       ;DO THIS LINE
CE46:C6 2A         321              DEC   BAS2L
```

```
CE48:30 0B    CE55  322             BMI   SCR80RET    ;=>DONE (HIT TOP)
CE4A:2C 1A C0       323             BIT   RDTEXT      ;ARE WE IN MIXEDMODE?
CE4D:30 EF    CE3E  324             BMI   SCR80       ;NO, DO FULL SCREEN
CE4F:A5 2A          325             LDA   BAS2L       ;IF 80, ONLY DO BOTTOM
CE51:C9 14          326             CMP   #20         ; FOUR (4) LINES OF WINDOW
CE53:B0 E9    CE3E  327             BCS   SCR80
CE55:                328 *
CE55:         CE55  329 SCR80RET    EQU   *
CE55:8D 0D C0       330             STA   SET80VID    ;DISPLAY IN 80-MODE
CE58:         CE58  331 SCRNRET     EQU   *           ;USED BY SCRN84
CE58:68             332             PLA               ;RESTORE
CE59:8D 7B 05       333             STA   OURCH       ; CH AND
CE5C:68             334             PLA               ;  CV
CE5D:8D FB 05       335             STA   OURCV
CE60:4C 51 CB       336             JMP   BASCALC     ;RETURN VIA BASCALC (UGH!)
CE63:               337 *
CE63:               338 *
CE63:         CE63  339 FORATE      EQU   *
CE63:08             340             PHP               ;DON'T ALLOW IRQ WHILE
CE64:78             341             SEI               ; SCREENHOLES ARE WRONG
CE65:A0 00          342             LDY   #0
CE67:84 2B          343             STY   BAS2H
CE69:8C 01 C0       344             STY   SET80COL
CE6C:2C 54 C0       345             BIT   TXTPAGE1
CE6F:B1 28          346 FORATE1     LDA   (BASL),Y
CE71:2C 55 C0       347             BIT   TXTPAGE2
CE74:20 A3 CE       348             JSR   DO48
CE77:2C 54 C0       349             BIT   TXTPAGE1
CE7A:B1 28          350             LDA   (BASL),Y
CE7C:20 A3 CE       351             JSR   DO48
CE7F:C0 28          352             CPY   #40
CE81:90 EC    CE6F  353             BCC   FORATE1
CE83:               354 *
CE83:20 91 CE       355             JSR   CLRHALF     ;CLEAR RIGHT HALF
CE86:2C 55 C0       356             BIT   TXTPAGE2    ; OF BOTH PAGES
CE89:20 91 CE       357             JSR   CLRHALF
CE8C:2C 54 C0       358             BIT   TXTPAGE1
CE8F:28             359             PLP               ;OK TO ALLOW IRQ NOW
CE90:60             360             RTS
CE91:               361 *
CE91:         CE91  362 CLRHALF     EQU   *
CE91:A0 14          363             LDY   #20
CE93:A9 A0          364             LDA   #' '
CE95:24 32          365             BIT   INVFLG      ;WHICH MODE?
CE97:30 02    CE9B  366             BMI   CLRHALF2    ;=>NORMAL
CE99:29 7F          367             AND   #$7F        ;INVERSE
CE9B:         CE9B  368 CLRHALF2    EQU   *
CE9B:91 28          369             STA   (BASL),Y    ;STUFF THE BLANK
CE9D:C8             370             INY
CE9E:C0 28          371             CPY   #40
CEA0:D0 F9    CE9B  372             BNE   CLRHALF2
CEA2:60             373             RTS
CEA3:               374 *
CEA3:48             375 DO48        PHA
CEA4:98             376             TYA
CEA5:4A             377             LSR   A
CEA6:A8             378             TAY
CEA7:68             379             PLA
CEA8:91 28          380             STA   (BASL),Y
CEAA:E6 2B          381             INC   BAS2H
CEAC:A4 2B          382             LDY   BAS2H
CEAE:60             383             RTS
CEAF:                14             INCLUDE SUBS3
CEAF:                2 ----------------------------------------
CEAF:                3 * NAME     : SETCH
CEAF:                4 * FUNCTION: SET OURCH AND CH
CEAF:                5 * INPUT    : AC=CH VALUE
CEAF:                6 * OUTPUT   : OURCH, CH MOD 40
CEAF:                7 * VOLATILE: NOTHING
CEAF:                8 * CALLS    : NOTHING
CEAF:                9 ----------------------------------------
CEAF:               10 *
CEAF:         CEAF  11 SETCH        EQU   *
CEAF:8D 7B 05       12              STA   OURCH       ;STUFF OURCH
CEB2:85 24          13              STA   CH          ;STUFF IN CASE WE'RE 40 MODE
CEB4:8D 7B 04       14              STA   OLDCH
CEB7:2C 1F C0       15              BIT   RD80VID     ;IN 80-MODE?
CEBA:10 1D    CED9  16              BPL   SETCHRTS    ;=>NO, DONE
CEBC:               17 *
CEBC:               18 * IF WE'RE NEAR THE END OF OUR
CEBC:               19 *  80COL LINE, MOVE CH UP.  IF NOT,
CEBC:               20 *  LEAVE CH PINNED AT ZERO...
CEBC:               21 *
CEBC:A9 00          22              LDA   #0          ;PIN CH AT ZERO
CEBE:85 24          23              STA   CH
CEC0:8D 7B 04       24              STA   OLDCH       ;REMEMBER THE SETTING
CEC3:A5 21          25              LDA   WNDWDTH     ;CHECK IF NEAR THE END
CEC5:38             26              SEC
CEC6:ED 7B 05       27              SBC   OURCH       ;GET ABS CH
CEC9:C9 08          28              CMP   #8          ;NEAR THE END?
CECB:B0 0C    CED9  29              BCS   SETCHRTS    ;=>NOPE
CECD:85 24          30              STA   CH          ;YES, MOVE CH UP NEAR RIGHT
```

```
CECF:A9 28          31              LDA   #40
CED1:38             32              SEC
CED2:E5 24          33              SBC   CH
CED4:85 24          34              STA   CH            ;BASIC WILL SEE THAT NOW
CED6:8D 7B 04       35              STA   OLDCH         ;REMEMBER THE SETTING
CED9:               36 *
CED9:         CED9  37 SETCHRTS     EQU   *
CED9:AD 7B 05       38              LDA   OURCH         ;RESTORE AC
CEDC:60             39              RTS
CEDD:               40 ----------------------------------------
CEDD:               41 * NAME     : INVERT
CEDD:               42 * FUNCTION: INVERT CHAR AT CH/CV
CEDD:               43 * INPUT    : NOTHING
CEDD:               44 * OUTPUT   : CHAR AT CH/CV INVERTED
CEDD:               45 * VOLATILE: NOTHING
CEDD:               46 * CALLS    : PICK, STORCHAR
CEDD:               47 ----------------------------------------
CEDD:               48 *
CEDD:         CEDD  49 INVERT     EQU   *
CEDD:48             50              PHA                 ;SAVE AC
CEDE:98             51              TYA                 ; AND Y
CEDF:48             52              PHA
CEE0:AC 7B 05       53              LDY   OURCH         ;GET CH
CEE3:20 01 CF       54              JSR   PICK          ;GET CHARACTER
CEE6:49 80          55              EOR   #$80          ;FLIP INVERSE/NORMAL
CEE8:2C 00 CF       56              BIT   SEV           ;PUT DIRECTLY BACK
CEEB:20 06 CF       57              JSR   SCREENIT      ; ONTO SCREEN
CEEE:68             58              PLA                 ;RESTORE Y
CEEF:A8             59              TAY                 ; AND AC
CEF0:68             60              PLA
CEF1:60             61              RTS
CEF2:               62 ----------------------------------------
CEF2:               63 * NAME     : STORCHAR
CEF2:               64 * FUNCTION: STORE A CHAR ON SCREEN
CEF2:               65 * INPUT    : AC=CHAR
CEF2:               66 *          : Y=CH POSITION
CEF2:               67 * OUTPUT   : CHAR ON SCREEN
CEF2:               68 * VOLATILE: NOTHING
CEF2:               69 * CALLS    : SCREENIT
CEF2:               70 ----------------------------------------
CEF2:               71 *
CEF2:         CEF2  72 STORCHAR     EQU   *
CEF2:48             73              PHA                 ;SAVE AC
CEF3:24 32          74              BIT   INVFLG        ;NORMAL OR INVERSE?
CEF5:30 02    CEF9  75              BMI   STOR2         ;=>NORMAL
CEF7:49 80          76              EOR   #$80          ;INVERSE
CEF9:         CEF9  77 STOR2        EQU   *
CEF9:2C 00 CF       78              BIT   SEV           ;V SET FOR STORE
CEFC:20 06 CF       79              JSR   SCREENIT      ;=>DO IT!
CEFF:68             80              PLA                 ;RESTORE AC
CF00:60             81 SEV          RTS
CF01:               82 ----------------------------------------
CF01:               83 * NAME     : PICK
CF01:               84 * FUNCTION: GET A CHAR FROM SCREEN
CF01:               85 * INPUT    : Y=CH POSITION
CF01:               86 * OUTPUT   : AC=CHARACTER
CF01:               87 * VOLATILE: NOTHING
CF01:               88 * CALLS    : SCREENIT
CF01:               89 ----------------------------------------
CF01:               90 *
CF01:         CF01  91 PICK         EQU   *
CF01:B8             92              CLV                 ;V CLEAR FOR PICK
CF02:20 06 CF       93              JSR   SCREENIT      ;DO IT!
CF05:60             94              RTS
CF06:               95 ----------------------------------------
CF06:               96 * NAME     : SCREENIT
CF06:               97 * FUNCTION: STORE OR PICK CHAR
CF06:               98 * INPUT    : V CLR FOR PICK
CF06:               99 *          : V SET FOR STORE
CF06:              100 *          : AC=CHAR FOR STORE
CF06:              101 *          : Y=CH POSITION
CF06:              102 * OUTPUT   : AC=CHAR (PICK)
CF06:              103 * VOLATILE: NOTHING
CF06:              104 * CALLS    : NOTHING
CF06:              105 ----------------------------------------
CF06:              106 *
CF06:         CF06 107 SCREENIT     EQU   *
CF06:84 1F         108              STY   YSAV1         ;SAVE Y
CF08:48            109              PHA                 ;SAVE CHARACTER IF STORING
CF09:              110 * AVOID CHANGING VFLAG VIA BIT!
CF09:AD 1F C0      111              LDA   RDBOVID       ;WHAT DISPLAY MODE?
CF0C:10 32    CF40 112              BPL   SCRN40        ;=>40-COL MODE
CF0E:              113 *
CF0E:              114 * 80-COLUMN MODE:
CF0E:              115 *
CF0E:A5 1F         116              LDA   YSAV1         ;GET CURSOR HORIZ
CF10:4A            117              LSR   A             ;DIVIDE BY TWO FOR PAGE
CF11:A8            118              TAY                 ;CH TO YREG
CF12:70 16    CF2A 119              BVS   STOR80        ;=>GONNA STORE THE CHAR
CF14:              120 *
CF14:              121 * 80-COL PICK:
```

```
CF14:                  122 *
CF14:08                123             PHP             ;LOCK INTERRUPTS WHILE
CF15:78                124             SEI             ; SCREENHOLES ARE WRONG
CF16:AD 55 CO          125             LDA   TXTPAGE2  ;ASSUME PAGE 2 (EVENS)
CF19:90 03   CF1E      126             BCC   SCRN2     ;=>IT IS
CF1B:AD 54 CO          127             LDA   TXTPAGE1  ;ODDS GO TO PAGE1
CF1E:          CF1E    128 SCRN2       EQU   *
CF1E:B1 28             129             LDA   (BASL),Y  ;PICK THE CHARACTER
CF20:A8                130             TAY             ;HOLD CHAR TEMPORARILY
CF21:AD 54 CO          131             LDA   TXTPAGE1  ;RESTORE PAGE1
CF24:28                132             PLP             ; AND ALLOW IRQ AGAIN
CF25:68                133             PLA             ;TRASH SAVED AC
CF26:98                134             TYA
CF27:48                135             PHA             ;MAKE CHAR GET RESTORED TO AC
CF28:50 24   CF4E      136             BVC   STPKEXIT  ;=>DONE (ALWAYS TAKEN)
CF2A:                  137 *
CF2A:          CF2A    138 STOR80      EQU   *
CF2A:68                139             PLA             ;RESTORE CHARACTER
CF2B:48                140             PHA             ;(LEAVE ON STACK)
CF2C:08                141             PHP             ;LOCK INTERRUPTS WHILE
CF2D:78                142             SEI             ; THE SCREENHOLES ARE WRONG
CF2E:48                143             PHA             ;HOLD THE CHAR TEMPORARILY
CF2F:AD 55 CO          144             LDA   TXTPAGE2  ;ASSUME PAGE2 (EVENS)
CF32:90 03   CF37      145             BCC   SCRN3     ;=>IT IS
CF34:AD 54 CO          146             LDA   TXTPAGE1  ;ODDS GO TO PAGE1
CF37:          CF37    147 SCRN3       EQU   *
CF37:68                148             PLA             ;GET CHAR TO BE STORED
CF38:91 28             149             STA   (BASL),Y  ;STUFF ONTO SCREEN
CF3A:AD 54 CO          150             LDA   TXTPAGE1  ;RESTORE PAGE1
CF3D:28                151             PLP             ; AND ALLOW IRQ AGAIN
CF3E:70 0E   CF4E      152             BVS   STPKEXIT  ;=>DONE (ALWAYS TAKEN)
CF40:                  153 *
CF40:                  154 * 40-COLUMN MODE:
CF40:                  155 *
CF40:          CF40    156 SCRN40      EQU   *
CF40:A4 1F             157             LDY   YSAV1     ;GET CURSOR HORIZ
CF42:70 06   CF4A      158             BVS   STOR40    ;=>STORE IT
CF44:68                159             PLA             ;TRASH SAVED CHAR
CF45:B1 28             160             LDA   (BASL),Y  ;PICK THE CHARACTER
CF47:48                161             PHA             ;SAVE CHAR FOR RESTORE
CF48:50 04   CF4E      162             BVC   STPKEXIT  ;DONE (ALWAYS TAKEN)
CF4A:                  163 *
CF4A:          CF4A    164 STOR40      EQU   *
CF4A:68                165             PLA             ;GET THE CHARACTER
CF4B:48                166             PHA             ;(LEAVE ON STACK)
CF4C:91 28             167             STA   (BASL),Y  ;STUFF ONTO SCREEN
CF4E:                  168 *
CF4E:          CF4E    169 STPKEXIT    EQU   *
CF4E:68                170             PLA             ;RESTORE AC
CF4F:A4 1F             171             LDY   YSAV1     ;RESTORE Y
CF51:60                172             RTS
CF52:                  173 ----------------------------------------
CF52:                  174 * NAME     : ESCON
CF52:                  175 * FUNCTION: TURN ON 'ESCAPE' CURSOR
CF52:                  176 * INPUT    : NONE
CF52:                  177 * OUTPUT   : 'CHAR'=ORIGINAL CHAR
CF52:                  178 * VOLATELE: NOTHING
CF52:                  179 * CALLS    : PICK,STORCHAR
CF52:                  180 ----------------------------------------
CF52:                  181 *
CF52:          CF52    182 ESCON       EQU   *
CF52:48                183             PHA             ;SAVE AC
CF53:98                184             TYA             ; AND Y
CF54:48                185             PHA
CF55:AC 7B 05          186             LDY   OURCH     ;GET CH
CF58:20 01 CF          187             JSR   PICK      ;GET ORIGINAL CHARACTER
CF5B:8D 7B 06          188             STA   CHAR      ; AND REMEMBER FOR ESCOFF
CF5E:29 80             189             AND   #$80      ;SAVE NORMAL/INVERSE BIT
CF60:49 AB             190             EOR   #$AB      ;MAKE IT AN INVERSE '+'
CF62:4C 6E CF          191             JMP   ESCRET    ;RETURN VIA SIMILAR CODE
CF65:                  192 ----------------------------------------
CF65:                  193 * NAME     : ESCOFF
CF65:                  194 * FUNCTION: TURN OFF 'ESCAPE' CURSOR
CF65:                  195 * INPUT    : 'CHAR'=ORIGINAL CHAR
CF65:                  196 * OUTPUT   : NONE
CF65:                  197 * VOLATILE: NOTHING
CF65:                  198 * CALLS    : STORCHAR
CF65:                  199 ----------------------------------------
CF65:                  200 *
CF65:          CF65    201 ESCOFF      EQU   *
CF65:48                202             PHA             ;SAVE AC
CF66:98                203             TYA             ; AND Y
CF67:48                204             PHA
CF68:AC 7B 05          205             LDY   OURCH     ;GET CH
CF6B:AD 7B 06          206             LDA   CHAR      ;GET ORIGINAL CHARACTER
CF6E:          CF6E    207 ESCRET      EQU   *         ;USED BY ESCON
CF6E:2C 00 CF          208             BIT   SEV       ; AND PUT IT BACK
CF71:20 06 CF          209             JSR   SCREENIT  ;  EXACTLY AS IT WAS
CF74:68                210             PLA             ;RESTORE Y
CF75:A8                211             TAY
CF76:68                212             PLA             ; AND AC
```

```
CF77:60            213             RTS
CF78:              214 ----------------------------------------
CF78:              215 * NAME     : COPYROM
CF78:              216 * FUNCTION: COPY F8 ROM TO LCARD
CF78:              217 * INPUT    : NOTHING
CF78:              218 * VOLATILE: X,Y
CF78:              219 * CALLS    : NOTHING
CF78:              220 ----------------------------------------
CF78:              221 *
CF78:       CF78   222 COPYROM     EQU   *
CF78:48            223             PHA                 ;SAVE AC
CF79:08            224             PHP                 ;ENSURE IRQ INHIBITED
CF7A:78            225             SEI                 ; WHILE COPYING ROM
CF7B:              226 *
CF7B:AD 11 C0      227             LDA   RDLCBNK2      ;GET BANK2
CF7E:48            228             PHA
CF7F:              229 *
CF7F:AE 12 C0      230             LDX   RDLCRAM       ; AND RAM FLAGS
CF82:AD 81 C0      231             LDA   $C081         ;SET READ-ROM
CF85:AD 81 C0      232             LDA   $C081         ; WRITE-RAM MODE
CF88:              233 *
CF88:A0 00         234             LDY   #0
CF8A:A9 F8         235             LDA   #$F8
CF8C:85 37         236             STA   CSWH          ;USE HOOK FOR MOVE
CF8E:A5 36         237             LDA   CSWL          ;PRESERVE LO BYTE
CF90:48            238             PHA
CF91:A9 00         239             LDA   #0
CF93:85 36         240             STA   CSWL
CF95:       CF95   241 COPYROM2    EQU   *             ;COPY ONLY PATCHED PAGES
CF95:B1 36         242             LDA   (CSWL),Y      ;MOVE THE ROM
CF97:91 36         243             STA   (CSWL),Y
CF99:C8            244             INY
CF9A:D0 F9   CF95  245             BNE   COPYROM2
CF9C:E6 37         246             INC   CSWH
CF9E:D0 F5   CF95  247             BNE   COPYROM2
CFA0:              248 *
CFA0:68            249             PLA                 ;RESTORE THE
CFA1:85 36         250             STA   CSWL          ; HOOK
CFA3:A9 C3         251             LDA   #<CNOO
CFA5:85 37         252             STA   CSWH
CFA7:              253 *
CFA7:68            254             PLA                 ;WHICH LC BANK?
CFA8:10 0F   CFB9  255             BPL   LCB1          ;=>BANK1
CFAA:8A            256             TXA                 ;RAM OR ROM READ?
CFAB:10 06   CFB3  257             BPL   LCB2ROM       ;=>ROM
CFAD:AD 80 C0      258             LDA   $C080         ;BANK2,RAM
CFB0:4C C5 CF      259             JMP   COPYRET
CFB3:AD 81 C0      260 LCB2ROM     LDA   $C081         ;BANK2,ROM
CFB6:4C C5 CF      261             JMP   COPYRET
CFB9:8A            262 LCB1        TXA                 ;RAM OR ROM READ?
CFBA:10 06   CFC2  263             BPL   LCB1ROM       ;=>ROM
CFBC:AD 88 C0      264             LDA   $C088         ;BANK1,RAM
CFBF:4C C5 CF      265             JMP   COPYRET
CFC2:AD 89 C0      266 LCB1ROM     LDA   $C089         ;BANK1,ROM
CFC5:              267 *
CFC5:       CFC5   268 COPYRET     EQU   *
CFC5:28            269             PLP                 ;RESTORE IRQ STATE NOW
CFC6:68            270             PLA                 ;   AND AC
CFC7:60            271             RTS
CFC8:              272 ----------------------------------------
CFC8:              273 * NAME     : PSETUP
CFC8:              274 * FUNCTION: SETUP ZP FOR PASCAL
CFC8:              275 * INPUT    : NONE
CFC8:              276 * OUTPUT   : NONE
CFC8:              277 * VOLATILE: AC
CFC8:              278 * CALLS    : NOTHING
CFC8:              279 ----------------------------------------
CFC8:              280 *
CFC8:       CFC8   281 PSETUP      EQU   *
CFC8:AD FB 04      282             LDA   MODE          ;TRANSPARENT MODE?
CFCB:29 01         283             AND   #M.TRANS
CFCD:D0 03   CFD2  284             BNE   PSETUP2       ;=>YES, TRUST WINDOW
CFCF:20 9B CD      285             JSR   FULL80        ;SET FULL 80COL WINDOW
CFD2:              286 *
CFD2:       CFD2   287 PSETUP2     EQU   *
CFD2:A9 FF         288             LDA   #255
CFD4:85 32         289             STA   INVFLG        ;ASSUME NORMAL MODE
CFD6:              290 *
CFD6:AD FB 04      291             LDA   MODE
CFD9:29 04         292             AND   #M.VMODE
CFDB:F0 02   CFDF  293             BEQ   PSETUPRET     ;=>IT'S NORMAL
CFDD:46 32         294             LSR   INVFLG        ;MAKE IT INVERSE
CFDF:              295 *
CFDF:       CFDF   296 PSETUPRET   EQU   *
CFDF:AD 7B 07      297             LDA   OLDBASL       ;SET UP BASE ADDRESS
CFE2:85 28         298             STA   BASL
CFE4:AD FB 07      299             LDA   OLDBASH
CFE7:85 29         300             STA   BASH
CFE9:60            301             RTS
CFEA:              302 ----------------------------------------
CFEA:              303 * NOTE: ENTRIES 6-7 OF THESE TABLES
```

```
CFEA:                 304 *  ARE NOT USED. THUS THERE ARE
CFEA:                 305 *  SOME OTHER VALUES STUFFED IN.
CFEA:                 306 *
CFEA: 28              307 F.TABLE      DFB   >F.CLREOP-1
CFEB: 42              308              DFB   >F.HOME-1
CFEC: 4C              309              DFB   >F.SCROLL-1
CFED: 7C              310              DFB   >F.CLREOL-1
CFEE: 9B              311              DFB   >F.CLEOLZ-1
CFEF: E9              312              DFB   >B.RESET-1   ;USE SAME RESET
CFF0: FF 01           313 PLUSMINUS1   DFB   -1,1         ;SCROLL USES THIS
CFF2: 89              314              DFB   >F.SETWND-1
CFF3:                 315 *
CFF3: E0              316 B.TABLE      DFB   >B.CLREOP-1
CFF4: EC              317              DFB   >B.HOME-1
CFF5: CC              318              DFB   >B.SCROLL-1
CFF6: D2              319              DFB   >B.CLREOL-1
CFF7: D8              320              DFB   >B.CLEOLZ-1
CFF8: E9              321              DFB   >B.RESET-1   ;USE SAME RESET
CFF9: 23 22           322 WNDTAB       DFB   WNDBTM,WNDTOP ;SCROLL USES THIS
CFFB: E6              323              DFB   >B.SETWND-1
CFFC: 00              324              DFB   0            ;AVOID CFFF PIPELINING
CFFD:          CFFD    15 ZZEND        EQU   *
```

80-Column Symbol Table, Sorted by Symbol

```
   3D A1H          3C A1L          3F A2H          3E A2L
   43 A4H          42 A4L       CE13 ATEFOR1    CE0A ATEFOR
 CA02 B.CANLIT   C9DF B.CHKCAN   C1D9 B.CLEOLZ   C1D3 B.CLREOL
 C1E1 B.CLREOP   C26E B.ESCFIX   C272 B.ESCFIX2  C27A B.ESCFIX3
 CA0A B.FIXCHR   C9F7 B.FLIP     C1A4 B.FUNCO    C10E B.FUNCNE
?C100 B.FUNC     C211 B.FUNC1    C107 B.FUNCNK   C20E B.GETCH
 C1ED B.HOME     C905 B.INPUT    CA24 B.INRET    C28B B.KEYIN
 C29C B.KEYIN2   C9C6 B.NOPICK   C11F B.OLDFUNC  C1EA B.RESET
 C234 B.RESETX   C1CD B.SCROLL   C221 B.SETWND2  C219 B.SETWNDX
 C1E7 B.SETWND   CFF3 B.TABLE    C1FF B.VECTOR     28 BAS2H
   2A BAS2L      CB54 BASCALCZ   CB51 BASCALC    CB7E BASCLC3
 CB97 BASCLCX      29 BASH       C317 BASICENT   C336 BASICENT2
 C803 BASICINIT  C305 BASICIN   ?C300 BASICINT   C307 BASICOUT
   28 BASL       CBC3 BELL2      C100 BFUNCPG    C831 BINIT1A
 C816 BINIT1     C850 BINIT2     C8F6 BINPUT     C8E2 BIORET
 C252 BLAST      C896 BOUT       C8CC BPNCTL     C8A1 BPRINT
?CBE2 BS40       CB5B BSCLC1A    C855 BSCLC1     CB6D BSCLC2
 CBEB BSDONE     C398 CO1        C3A3 CO3        C874 C8B2
 C87E C8B3       C890 C8B4       C866 C8BASIC    07F8 C8SLOT
   24 CH         067B CHAR       C85D CLEARIT    C181 CLEOL2
 C12D CLEOP1     C000 CLR80COL   C00C CLR80VID   C00E CLRALTCHAR
 CE9B CLRHALF2   CE91 CLRHALF    C300 CNOO       CFC5 COPYRET
 CF95 COPYROM2   CF78 COPYROM      37 CSWH       C36 CSWL
 CC78 CTLADH     CC5F CTLADL     CB99 CTLCHAR    CBAB CTLCHARX
 CBAE CTLGO      CBB2 CTLRET     CBB6 CTLXFER      25 CV
 C261 DIAGS      CEA3 DO48       C929 ESC1       C92B ESC2
 C935 ESC3       C918 ESCAPING   C983 ESCCHAR    C280 ESCIN
 C960 ESCNONE    0011 ESCNUM     CF65 ESCOFF     CF52 ESCON
 C284 ESCOUT     CF6E ESCRET     C945 ESCSPEC    C954 ESCSPEC2
 C963 ESCSPEC3   C972 ESCTAB    ?FBC1 F.BASCALC  C19C F.CLEOLZ
 C17D F.CLREOL   C129 F.CLREOP   C1A1 F.GORET    C143 F.HOME
 C2F1 F.RET1     C2EB F.RETURN   C14D F.SCROLL   C18A F.SETWND
 CFEA F.TABLE    FC22 F.VTAB     FC24 F.VTABZ    FBB3 F8VERSION
 CE6F FORATE1    CE63 FORATE     CD9B FULL80     FD29 FUNCEXIT
 CE22 GET84      CB1B GETK2      CB15 GETKEY     CA27 GETPRIOR
 CAAF GETY       C22E GOBACK       06 GOODF8     C27D GORETN
 C2B5 GOTKEY     CA49 GPX        C813 HANG       C2CC IK1
 C2D5 IK2A       C2CE IK2        C2DB IK3        CEDD INVERT
   32 INVFLG     FF58 IORTS      C348 JBASINIT   C34B JPINIT
 C351 JPREAD     C35D JPSTAT     C357 JPWRITE    C010 KBDSTRB
 C000 KBD        C8B4 KBDWAIT    C2EA KDRET     ?C2E6 KDRETN
 C2E9 KDRETY     C2C6 KEYDLY       39 KSWH         38 KSWL
 CFB9 LCB1       CFC2 LCB1ROM    CFB3 LCB2ROM      40 M.BINPUT
   80 M.ESCR       08 M.GOXY       01 M.IRQ        10 M.LIT
   02 M.PAS1.O     20 M.PASCAL     01 M.TRANS      04 M.VMODE
 04FB MODE       C378 MOVEC2M    C380 MOVELOOP   C3AC MOVERET
 C37E MOVESTRT   C363 MOVE       CCDD MSCRL0     CCE1 MSCRL1
 CCF9 MSCRL2     CD02 MSCRLRET   CD10 MSCRLRTS   C9B7 NOESC
 C1C5 NOI        C8C0 NOWAIT     C38A NXTA1      07FB OLDBASH
 077B OLDBASL    047B OLDCH      CD09 ONEMORE    057B OURCH
 05FB OURCV      CF01 PICK       CA62 PIGOOD     CA4A PINIT1.O
 CA51 PINIT2     CA4F PINIT      CFF0 PLUSMINUS1 CA74 PREAD
 CA8A PREADRET2  CFD2 PSETUP2    CFC8 PSETUP     CFDF PSETUPRET
 C99E PSTATUS2   C994 PSTATUS    C9B0 PSTATUS3   C9B4 PSTATUS4
?CA9E PWRITE2    CACB PWRITE3    CABE PWRITE     CAEB PWRITE4
 CB0F PWRITERET  CB09 PWWRAP     CDC0 QUIT2      CDAA QUIT
 C018 RD80COL    C01F RD80VID    C003 RDCARDRAM ?FD0C RDKEY
 C011 RDLCBNK2   C012 RDLCRAM    C002 RDMAINRAM  C01C RDPAGE2
 C013 RDRAMRD    C014 RDRAMWRT   C01A RDTEXT    ?C019 RDVBLBAR
 C264 RESETRET     4F RNDH         4E RNDL       CE01 SCR40RET
 CDEA SCR40      CE3E SCR80      CE55 SCR80RET   CF06 SCREENIT
 C153 SCRL1      C169 SCRL2      C172 SCRL3      CCD1 SCRLSUB
 CF1E SCRN2      CF37 SCRN3      CF40 SCRN40     CE32 SCRN48
 CDDB SCRN84     CE58 SCRNRET    CCAE SCROLL1    CCBB SCROLL2
 CCAA SCROLLDN   CCA4 SCROLLUP   C001 SET80COL   C00D SET80VID
 C00F SETALTCHAR C009 SETALTZP   C3EB SETC8      CEAF SETCH
 CED9 SETCHRTS  ?C007 SETINTCXROM FE89 SETKBD    C00B SETSLOTC3ROM
 C008 SETSTDZP   FE93 SETVID     CF00 SEV        FC75 SNIFFIRQ
 C030 SPKR       CADC STARTXY    CB4B STAY80     CB4D STAY80
 CEF9 STOR2      CF4A STOR40     CF2A STOR80     CEF2 STORCHAR
 CF4E STPKEXIT   CC59 STUFFINV   0478 TEMP1        00 TEST
 CB24 TESTCARD  ?CB4E TESTFAIL   C054 TXTPAGE1   C055 TXTPAGE2
 CBCF WAIT       CBD0 WAIT2      CBD1 WAIT3        23 WNDBTM
   20 WNDLFT     CFF9 WNDTAB       22 WNDTOP       21 WNDWDTH
 C005 WRCARDRAM  C004 WRMAINRAM  CBBC X.BELL     CBDB X.BS
 C2F6 X.CLEOL2   C2F4 X.CLEOLZ   CC0C X.CRRET    CBEC X.CR
```

```
 CBFD X.CRPAS        CD64 X.DC1B         CD76 X.DC1RTS       CD59 X.DC1
 CD88 X.DC2B         CD9A X.DC2RET       CD77 X.DC2          CC0D X.EM
 CD42 X.FF           CC33 X.FSRET        CC26 X.FS           CD48 X.GS
 CD54 X.GS2          CD4E X.GSEOLZ       CC91 X.LF           CC9E X.LF2
 CD20 X.LFRET        CD90 X.NAK          CD9A X.NAKRET       CD11 X.SCRLRET
 CD17 X.SCRLRET2     CC52 X.SI           CC49 X.SO          ?CC1D X.SUB80
 CC1F X.SUBLP        CC1A X.SUB          CC34 X.US           CC40 X.VS1
 CC45 X.VS2          CC48 X.VSRET        CD23 X.VT           CD2C X.VTLOOP
 CD32 X.VTNEXT       06FB XCOORD         C3B0 XFER           C3CD XFERAZP
 C3C5 XFERC2M        C3DC XFERSZP          1F YSAV1           ?  02 ZSPAREC2
?CFFD ZZEND
** SUCCESSFUL ASSEMBLY : = NO ERRORS
** ASSEMBLER CREATED ON 05-JAN-82 000004
** TOTAL LINES ASSEMBLED  2419
** FREE SPACE PAGE COUNT   49
    2  EQUATES
    3  BFUNC
    4  C3SPACE
    5  C8SPACE
    6  BPRINT
    7  BINPUT
    8  PINIT
    9  PREAD
   10  PWRITE
   11  SUBS1
   12  SUBS2
   13  SUBS3
```

80-Column Symbol Table, Sorted by Address

```
   00 TEST          01 M.TRANS       01 M.IRQ         02 M.PAS1.0
?  02 ZSPAREC2      04 M.VMODE       06 GOODF8        08 M.GOXY
   10 M.LIT      0011 ESCNUM         1F YSAV1         20 M.PASCAL
   20 WNDLFT        21 WNDWDTH       22 WNDTOP        23 WNDBTM
   24 CH            25 CV            28 BASL          29 BASH
   2A BAS2L         2B BAS2H         32 INVFLG        36 CSWL
   37 CSWH          38 KSWL          39 KSWH          3C A1L
   3D A1H           3E A2L           3F A2H           40 M.BINPUT
   42 A4L           43 A4H           4E RNDL          4F RNDH
   B0 M.ESCR     0478 TEMP1       047B OLDCH       04FB MODE
057B OURCH       05FB OURCV       067B CHAR        06FB XCOORD
077B OLDBASL     07F8 C8SLOT      07FB OLDBASH     C000 CLR80COL
C000 KBD         C001 SET80COL    C002 RDMAINRAM   C003 RDCARDRAM
C004 WRMAINRAM   C005 WRCARDRAM ?C007 SETINTCXROM C008 SETSTDZP
C009 SETALTZP    C00B SETSLOTC3ROM C00C CLR80VID   C00D SET80VID
C00E CLRALTCHAR  C00F SETALTCHAR  C010 KBDSTRB    C011 RDLCBNK2
C012 RDLCRAM     C013 RDRAMRD     C014 RDRAMWRT    C018 RD80COL
?C019 RDVBLBAR   C01A RDTEXT      C01C RDPAGE2     C01F RD80VID
C030 SPKR        C054 TXTPAGE1    C055 TXTPAGE2 ?C100 B.FUNC
C100 BFUNCPG     C107 B.FUNCNK    C10E B.FUNCNE    C11F B.OLDFUNC
C129 F.CLREOP    C12D CLEOP1      C143 F.HOME      C14D F.SCROLL
C153 SCRL1       C169 SCRL2       C172 SCRL3       C17D F.CLREOL
C181 CLEOL2      C18A F.SETWND    C19C F.CLEOLZ    C1A1 F.GORET
C1A4 B.FUNC0     C1C5 NOI         C1CD B.SCROLL    C1D3 B.CLREOL
C1D9 B.CLEOLZ    C1E1 B.CLREOP    C1E7 B.SETWND    C1EA B.RESET
C1ED B.HOME      C1FF B.VECTOR    C20E B.GETCH     C211 B.FUNC1
C219 B.SETWNDX   C221 B.SETWND2   C22E GOBACK      C234 B.RESETX
C252 BLAST       C261 DIAGS       C264 RESETRET    C26E B.ESCFIX
C272 B.ESCFIX2   C27A B.ESCFIX3   C27D GORETN      C2B0 ESCIN
C2B4 ESCOUT      C288 B.KEYIN     C29C B.KEYIN2    C2B5 GOTKEY
C2C6 KEYDLY      C2CC IK1         C2CE IK2         C2D5 IK2A
C2DB IK3      ?C2E6 KDRETN        C2E9 KDRETY      C2EA KDRET
C2EB F.RETURN    C2F1 F.RET1      C2F4 X.CLEOLZ    C2F6 X.CLEOL2
C300 CNOO     ?C300 BASICINT      C305 BASICIN     C307 BASICOUT
C317 BASICENT    C336 BASICENT2   C348 JBASINIT    C34B JPINIT
C351 JPREAD      C357 JPWRITE     C35D JPSTAT      C363 MOVE
C378 MOVEC2M     C37E MOVESTRT    C380 MOVELOOP    C38A NXTA1
C398 CO1         C3A3 CO3         C3AC MOVERET     C3B0 XFER
C3C5 XFERC2M     C3CD XFERAZP     C3DC XFERSZP     C3EB SETC8
C803 BASICINIT   C813 HANG        C816 BINIT1      C831 BINIT1A
C850 BINIT2      C85D CLEARIT     C866 C8BASIC     C874 C8B2
C87E C8B3        C890 C8B4        C896 BOUT        C8A1 BPRINT
C8B4 KBDWAIT     C8C0 NOWAIT      C8CC BPNCTL      C8E2 BIORET
C8F6 BINPUT      C905 B.INPUT     C918 ESCAPING    C929 ESC1
C92B ESC2        C935 ESC3        C945 ESCSPEC     C954 ESCSPEC2
C960 ESCNONE     C963 ESCSPEC3    C972 ESCTAB      C9B3 ESCCHAR
C994 PSTATUS     C99E PSTATUS2    C9B0 PSTATUS3    C9B4 PSTATUS4
C9B7 NOESC       C9C6 B.NOPICK    C9DF B.CHKCAN    C9F7 B.FLIP
CA02 B.CANLIT    CA0A B.FIXCHR    CA24 B.INRET     CA27 GETPRIOR
CA49 GPX         CA4A PINIT1.0    CA4F PINIT       CA51 PINIT2
CA62 PIGOOD      CA74 PREAD       CA8A PREADRET2   CABE PWRITE
?CA9E PWRITE2    CAAF GETY        CACB PWRITE3     CADC STARTXY
CAEB PWRITE4     CB09 PWWRAP      CB0F PWRITERET   CB15 GETKEY
CB1B GETK2       CB24 TESTCARD    CB48 STAY2       CB4D STAY80
?CB4E TESTFAIL   CB51 BASCALC     CB54 BASCALCZ    CB55 BSCLC1
CB5B BSCLC1A     CB6D BSCLC2      CB7E BASCLC3     CB97 BASCLCX
CB99 CTLCHAR     CBAB CTLCHARX    CBAE CTLGO       CBB2 CTLRET
CBB6 CTLXFER     CBBC X.BELL      CBC3 BELL2       CBCF WAIT
CBD0 WAIT2       CBD1 WAIT3       CBDB X.BS     ?CBE2 BS40
CBEB BSDONE      CBEC X.CR        CBFD X.CRPAS     CC0C X.CRRET
CC0D X.EM        CC1A X.SUB    ?CC1D X.SUB80       CC1F X.SUBLP
CC26 X.FS        CC33 X.FSRET     CC34 X.US        CC40 X.US1
CC45 X.US2       CC48 X.USRET     CC49 X.SO        CC52 X.SI
CC59 STUFFINV    CC5F CTLADL      CC78 CTLADH      CC91 X.LF
CC9E X.LF2       CCA4 SCROLLUP    CCAA SCROLLDN    CCAE SCROLL1
CCB8 SCROLL2     CCD1 SCRLSUB     CCDD MSCRL0      CCE1 MSCRL1
CCF9 MSCRL2      CD02 MSCRLRET    CD09 ONEMORE     CD10 MSCRLRTS
CD11 X.SCRLRET   CD17 X.SCRLRET2  CD20 X.LFRET     CD23 X.VT
CD2C X.VTLOOP    CD32 X.VTNEXT    CD42 X.FF        CD48 X.GS
CD4E X.GSEOLZ    CD54 X.GS2       CD59 X.DC1       CD64 X.DC1B
CD76 X.DC1RTS    CD77 X.DC2       CD88 X.DC2B      CD90 X.NAK
CD9A X.NAKRET    CD9A X.DC2RET    CD9B FULL80      CDAA QUIT
CDC0 QUIT2       CDDB SCRN84      CDEA SCR40       CE01 SCR40RET
CE0A ATEFOR      CE13 ATEFOR1     CE22 GET84       CE32 SCRN48
CE3E SCR80       CE55 SCR80RET    CE58 SCRNRET     CE63 FORATE
CE6F FORATE1     CE91 CLRHALF     CE9B CLRHALF2    CEA3 D048
```

```
 CEAF SETCH          CED9 SETCHRTS       CEDD INVERT         CEF2 STORCHAR
 CEF9 STOR2          CF00 SEV            CF01 PICK           CF06 SCREENIT
 CF1E SCRN2          CF2A STOR80         CF37 SCRN3          CF40 SCRN40
 CF4A STOR40         CF4E STPKEXIT       CF52 ESCON          CF65 ESCOFF
 CF6E ESCRET         CF78 COPYROM        CF95 COPYROM2       CFB3 LCB2ROM
 CFB9 LCB1           CFC2 LCB1ROM        CFC5 COPYRET        CFC8 PSETUP
 CFD2 PSETUP2        CFDF PSETUPRET      CFEA F.TABLE        CFF0 PLUSMINUS1
 CFF3 B.TABLE        CFF9 WNDTAB        ?CFFD ZZEND          FBB3 F8VERSION
?FBC1 F.BASCALC      FC22 F.VTAB         FC24 F.VTABZ        FC75 SNIFFIRQ
?FDOC RDKEY          FD29 FUNCEXIT       FE89 SETKBD         FE93 SETVID
 FF58 IORTS
** SUCCESSFUL ASSEMBLY := NO ERRORS
** ASSEMBLER CREATED ON 05-JAN-82 000004
** TOTAL LINES ASSEMBLED  2421
** FREE SPACE PAGE COUNT   49
    2  EQUATES
    3  BFUNC
    4  C3SPACE
    5  C8SPACE
    6  BPRINT
    7  BINPUT
    8  PINIT
    9  PREAD
   10  PWRITE
   11  SUBS1
   12  SUBS2
   13  SUBS3
```

apple computer

20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010
TLX 171576

031-0357-A