

```

CED4:      182 *
CED4:      183 PSETUP EQU *
CED4:20 71 CD 184 JSR FULL80 ;SET FULL 80COL WINDOW
CED7:A9 FF 185 LDA #255
CED9:85 32 186 STA INVFLG ;ASSUME NORMAL MODE
CEDB:      187 *
CEDB:AD FB 04 188 LDA MODE
CEDE:29 04 189 AND #M.VMODE
CEEO:F0 02 CEE4 190 BEQ PSETUPRET ;=>IT'S NORMAL
CEE2:46 32 191 LSR INVFLG ;MAKE IT INVERSE
CEE4:      192 *
CEE4:      193 PSETUPRET EQU *
CEE4:AD 7B 07 194 LDA OLDBASL ;SET UP BASE ADDRESS
CEE7:85 28 195 STA BASL
CEE9:AD FB 07 196 LDA OLDBASH
CEEC:85 29 197 STA BASH
CEEE:AD FB 05 198 LDA OURCV ;get user's cursor vertical
CEF1:85 25 199 STA CV ;and set it up
CEF3:60 200 RTS
CEF4:      201 *****
CEF4:      202 *
CEF4:      203 * COPYROM is called when the video firmware is
CEF4:      204 * initialized. If the language card is switched
CEF4:      205 * in for reading, it copies the F8 ROM to the
CEF4:      206 * language card and restores the state of the
CEF4:      207 * language card.
CEF4:      208 *
CEF4:2C 12 CO 209 COPYROM BIT RDLGRAM ;is the LC switched in?
CEF7:10 3D CF36 210 BPL ROMOK ;=>no, do nothing
CEF9:A9 06 211 LDA #GOODF8 ;yes, check $F8 RAM
CEFB:CD B3 FB 212 CMP F8VERSION ;does it match?
CEFE:F0 36 CF36 213 BEQ ROMOK ;=> assum ROM is there
CF00:A2 03 214 LDX #3 ;indicate bank 2, RAM write enabled
CF02:2C 11 CO 215 BIT RDLGBNK2 ;is it bank 2?
CF05:30 02 CF09 216 BMI BANK2 ;=>yes, we were right
CF07:A2 0B 217 LDX #B ;no, bank 1, RAM write enabled
CF09:8D B3 FB 218 BANK2 STA F8VERSION ;write to see if LC is
CF0C:2C 80 CO 219 BIT $C080 ;write protected (read RAM)
CF0F:AD B3 FB 220 LDA F8VERSION ;did it change?
CF12:C9 06 221 CMP #GOODF8
CF14:F0 01 CF17 222 BEQ WRITENBL ;=>yes, write enabled
CF16:E8 223 INX ;else indicate write protect
CF17:2C 81 CO 224 WRITENBL BIT $C081 ;read ROM, write RAM
CF1A:2C 81 CO 225 BIT $C081 ;twice is nice
CF1D:A0 00 226 LDY #0 ;now copy ROM to RAM
CF1F:A9 F8 227 LDA #$F8
CF21:85 37 228 STA CSMH ;hooks set later
CF23:84 36 229 STY CSWL
CF25:B1 36 230 COPYROM2 LDA (CSWL),Y ;get a byte
CF27:91 36 231 STA (CSWL),Y ;and move it
CF29:C8 232 INY
CF2A:DO F9 CF25 233 BNE COPYROM2
CF2C:E6 37 234 INC CSMH ;next page
CF2E:DO F5 CF25 235 BNE COPYROM2 ;finish copy
CF30:BD 80 CO 236 LDA $C080,x ;read RAM
CF33:BD 80 CO 237 LDA $C080,x
CF36:60 238 ROMOK RTS ;done with ROM copy

```

```

0000:      0000 1 TEST EQU 0
0000:      2 LST On,A,V
0000:      0001 3 IRQTEST EQU 1
0000:      4 MSB ON ;SET THEM HIBITS
0000:      0000 5 DO TEST
0000:      6 F8ORG EQU $1800
0000:      7 IOADR EQU $2000 ;For setting PR# hooks
0000:      8 C1ORG EQU $2100
0000:      9 C3ORG EQU $2300
0000:     10 C8ORG EQU $2800
0000:     11 ELSE
0000:     F800 12 F8ORG EQU $F800
0000:     C100 13 C1ORG EQU $C100
0000:     C300 14 C3ORG EQU $C300
0000:     C800 15 C8ORG EQU $C800
0000:     16 FIN
0000:      2 *****
0000:      3 *
0000:      4 * APPLE II
0000:      5 * MONITOR II
0000:      6 *
0000:      7 * COPYRIGHT 1978, 1981, 1984 BY
0000:      8 * APPLE COMPUTER, INC.
0000:      9 *
0000:     10 * ALL RIGHTS RESERVED
0000:     11 *
0000:     12 * S. WOZNIAC 1977
0000:     13 * A. BAUM 1977
0000:     14 * JOHN A NOV 1978
0000:     15 * R. AURICCHIO SEP 1981
0000:     16 * E. BEERNINK 1984
0000:     17 *
0000:     0001 18 APPLE2F EQU 1 ;COND ASSM/RAA0981
0000:     19 *
0000:     20 *****
F800:     F800 21 ORG F8ORG
F800:     2000 22 OBJ $2000
F800:     23 *****
F800:     24 *
F800:     25 * Zero Page Equates
F800:     26 *
F800:     0000 27 LOCO EQU $00 ;vector for autost from disk
F800:     0001 28 LOCL EQU $01
F800:     0020 29 WNDLFT EQU $20 ;left edge of text window
F800:     0021 30 WNDWDT EQU $21 ;width of text window
F800:     0022 31 WNDTOP EQU $22 ;top of text window
F800:     0023 32 WNDBTM EQU $23 ;bottom+1 of text window
F800:     0024 33 CH EQU $24 ;cursor horizontal position
F800:     0025 34 CV EQU $25 ;cursor vertical position
F800:     0026 35 GBASL EQU $26 ;lo-res graphics base addr.
F800:     0027 36 GBASH EQU $27
F800:     0028 37 BASL EQU $28 ;text base address

```