```
***************************
*                         *
*        APPLE II         *
*     SYSTEM MONITOR      *
*                         *
*    COPYRIGHT 1977 BY    *
*   APPLE COMPUTER, INC.  *
*                         *
*   ALL RIGHTS RESERVED   *
*                         *
*       S. WOZNIAK        *
*        A. BAUM          *
*                         *
***************************
; TITLE "APPLE II SYSTEM MONITOR"
LOC0     EQU   $00
LOC1     EQU   $01
WNDLFT   EQU   $20
WNDWDTH  EQU   $21
WNDTOP   EQU   $22
WNDBTM   EQU   $23
CH       EQU   $24
CV       EQU   $25
GBASL    EQU   $26
GBASH    EQU   $27
BASL     EQU   $28
BASH     EQU   $29
BAS2L    EQU   $2A
BAS2H    EQU   $2B
H2       EQU   $2C
LMNEM    EQU   $2C
RTNL     EQU   $2C
V2       EQU   $2D
RMNEM    EQU   $2D
RTNH     EQU   $2D
MASK     EQU   $2E
CHKSUM   EQU   $2E
FORMAT   EQU   $2E
LASTIN   EQU   $2F
LENGTH   EQU   $2F
SIGN     EQU   $2F
COLOR    EQU   $30
MODE     EQU   $31
INVFLG   EQU   $32
PROMPT   EQU   $33
YSAV     EQU   $34
YSAV1    EQU   $35
CSWL     EQU   $36
CSWH     EQU   $37
KSWL     EQU   $38
KSWH     EQU   $39
PCL      EQU   $3A
PCH      EQU   $3B
XQT      EQU   $3C
```

```
                A1L      EQU   $3C
                A1H      EQU   $3D
                A2L      EQU   $3E
                A2H      EQU   $3F
                A3L      EQU   $40
                A3H      EQU   $41
                A4L      EQU   $42
                A4H      EQU   $43
                A5L      EQU   $44
                A5H      EQU   $45
                ACC      EQU   $45
                XREG     EQU   $46
                YREG     EQU   $47
                STATUS   EQU   $48
                SPNT     EQU   $49
                RNDL     EQU   $4E
                RNDH     EQU   $4F
                ACL      EQU   $50
                ACH      EQU   $51
                XTNDL    EQU   $52
                XTNDH    EQU   $53
                AUXL     EQU   $54
                AUXH     EQU   $55
                PICK     EQU   $95
                IN       EQU   $0200
                USRADR   EQU   $03F8
                NMI      EQU   $03FB
                IRQLOC   EQU   $03FE
                IOADR    EQU   $C000
                KBD      EQU   $C000
                KBDSTRB  EQU   $C010
                TAPEOUT  EQU   $C020
                SPKR     EQU   $C030
                TXTCLR   EQU   $C050
                TXTSET   EQU   $C051
                MIXCLR   EQU   $C052
                MIXSET   EQU   $C053
                LOWSCR   EQU   $C054
                HISCR    EQU   $C055
                LORES    EQU   $C056
                HIRES    EQU   $C057
                TAPEIN   EQU   $C060
                PADDL0   EQU   $C064
                PTRIG    EQU   $C070
                BASIC    EQU   $E000
                BASIC2   EQU   $E003
                         ORG   $F800       ;ROM START ADDRESS
F800: 4A        PLOT     LSR               ;Y-COORD/2
F801: 08                 PHP               ;SAVE LSB IN CARRY
F802: 20 47 F8           JSR   GBASCALC    ;CALC BASE ADR IN GBASL,H
F805: 28                 PLP               ;RESTORE LSB FROM CARRY
F806: A9 0F              LDA   #$0F        ;MASK $0F IF EVEN
F808: 90 02              BCC   RTMASK
F80A: 69 E0              ADC   #$E0        ;MASK $F0 IF ODD
F80C: 85 2E     RTMASK   STA   MASK
```

```
F80E: B1 26     PLOT1    LDA   (GBASL),Y  ;DATA
F810: 45 30              EOR   COLOR      ; EOR COLOR
F812: 25 2E              AND   MASK       ;  AND MASK
F814: 51 26              EOR   (GBASL),Y  ;   EOR DATA
F816: 91 26              STA   (GBASL),Y  ;    TO DATA
F818: 60                 RTS
F819: 20 00 F8  HLINE    JSR   PLOT       ;PLOT SQUARE
F81C: C4 2C     HLINE1   CPY   H2         ;DONE?
F81E: B0 11              BCS   RTS1       ; YES, RETURN
F820: C8                 INY              ; NO, INC INDEX (X-COORD)
F821: 20 0E F8           JSR   PLOT1      ;PLOT NEXT SQUARE
F824: 90 F6              BCC   HLINE1     ;ALWAYS TAKEN
F826: 69 01     VLINEZ   ADC   #$01       ;NEXT Y-COORD
F828: 48        VLINE    PHA              ; SAVE ON STACK
F829: 20 00 F8           JSR   PLOT       ; PLOT SQUARE
F82C: 68                 PLA
F82D: C5 2D              CMP   V2         ;DONE?
F82F: 90 F5              BCC   VLINEZ     ; NO, LOOP
F831: 60        RTS1     RTS
F832: A0 2F     CLRSCR   LDY   #$2F       ;MAX Y, FULL SCRN CLR
F834: D0 02              BNE   CLRSC2     ;ALWAYS TAKEN
F836: A0 27     CLRTOP   LDY   #$27       ;MAX Y, TOP SCREEN CLR
F838: 84 2D     CLRSC2   STY   V2         ;STORE AS BOTTOM COORD
                                          ; FOR VLINE CALLS
F83A: A0 27              LDY   #$27       ;RIGHTMOST X-COORD (COLUMN)
F83C: A9 00     CLRSC3   LDA   #$00       ;TOP COORD FOR VLINE CALLS
F83E: 85 30              STA   COLOR      ;CLEAR COLOR (BLACK)
F840: 20 28 F8           JSR   VLINE      ;DRAW VLINE
F843: 88                 DEY              ;NEXT LEFTMOST X-COORD
F844: 10 F6              BPL   CLRSC3     ;LOOP UNTIL DONE
F846: 60                 RTS
F847: 48        GBASCALC PHA              ;FOR INPUT 000DEFGH
F848: 4A                 LSR
F849: 29 03              AND   #$03
F84B: 09 04              ORA   #$04       ;  GENERATE GBASH=000001FG
F84D: 85 27              STA   GBASH
F84F: 68                 PLA              ;  AND GBASL=HDEDE000
F850: 29 18              AND   #$18
F852: 90 02              BCC   GBCALC
F854: 69 7F              ADC   #$7F
F856: 85 26     GBCALC   STA   GBASL
F858: 0A                 ASL
F859: 0A                 ASL
F85A: 05 26              ORA   GBASL
F85C: 85 26              STA   GBASL
F85E: 60                 RTS
F85F: A5 30     NXTCOL   LDA   COLOR      ;INCREMENT COLOR BY 3
F861: 18                 CLC
F862: 69 03              ADC   #$03
F864: 29 0F     SETCOL   AND   #$0F       ;SETS COLOR=17*A MOD 16
F866: 85 30              STA   COLOR
F868: 0A                 ASL              ;BOTH HALF BYTES OF COLOR EQUAL
F869: 0A                 ASL
F86A: 0A                 ASL
F86B: 0A                 ASL
```

```
F86C: 05 30              ORA   COLOR
F86E: 85 30              STA   COLOR
F870: 60                 RTS
F871: 4A        SCRN     LSR               ;READ SCREEN Y-COORD/2
F872: 08                 PHP               ;SAVE LSB (CARRY)
F873: 20 47 F8           JSR   GBASCALC    ;CALC BASE ADDRESS
F876: B1 26              LDA   (GBASL),Y   ;GET BYTE
F878: 28                 PLP               ;RESTORE LSB FROM CARRY
F879: 90 04     SCRN2    BCC   RTMSKZ      ;IF EVEN, USE LO H
F87B: 4A                 LSR
F87C: 4A                 LSR
F87D: 4A                 LSR               ;SHIFT HIGH HALF BYTE DOWN
F87E: 4A                 LSR
F87F: 29 0F     RTMSKZ   AND   #$0F        ;MASK 4-BITS
F881: 60                 RTS
F882: A6 3A     INSDS1   LDX   PCL         ;PRINT PCL,H
F884: A4 3B              LDY   PCH
F886: 20 96 FD           JSR   PRYX2
F889: 20 48 F9           JSR   PRBLNK      ;FOLLOWED BY A BLANK
F88C: A1 3A              LDA   (PCL,X)     ;GET OP CODE
F88E: A8        INSDS2   TAY
F88F: 4A                 LSR               ;EVEN/ODD TEST
F890: 90 09              BCC   IEVEN
F892: 6A                 ROR               ;BIT 1 TEST
F893: B0 10              BCS   ERR         ;XXXXXX11 INVALID OP
F895: C9 A2              CMP   #$A2
F897: F0 0C              BEQ   ERR         ;OPCODE $89 INVALID
F899: 29 87              AND   #$87        ;MASK BITS
F89B: 4A        IEVEN    LSR               ;LSB INTO CARRY FOR L/R TEST
F89C: AA                 TAX
F89D: BD 62 F9           LDA   FMT1,X      ;GET FORMAT INDEX BYTE
F8A0: 20 79 F8           JSR   SCRN2       ;R/L H-BYTE ON CARRY
F8A3: D0 04              BNE   GETFMT
F8A5: A0 80     ERR      LDY   #$80        ;SUBSTITUTE $80 FOR INVALID OPS
F8A7: A9 00              LDA   #$00        ;SET PRINT FORMAT INDEX TO 0
F8A9: AA        GETFMT   TAX
F8AA: BD A6 F9           LDA   FMT2,X      ;INDEX INTO PRINT FORMAT TABLE
F8AD: 85 2E              STA   FORMAT      ;SAVE FOR ADR FIELD FORMATTING
F8AF: 29 03              AND   #$03        ;MASK FOR 2-BIT LENGTH
                                           ; (P=1 BYTE, 1=2 BYTE, 2=3 BYTE)
F8B1: 85 2F              STA   LENGTH
F8B3: 98                 TYA               ;OPCODE
F8B4: 29 8F              AND   #$8F        ;MASK FOR 1XXX1010 TEST
F8B6: AA                 TAX               ; SAVE IT
F8B7: 98                 TYA               ;OPCODE TO A AGAIN
F8B8: A0 03              LDY   #$03
F8BA: E0 8A              CPX   #$8A
F8BC: F0 0B              BEQ   MNNDX3
F8BE: 4A        MNNDX1   LSR
F8BF: 90 08              BCC   MNNDX3      ;FORM INDEX INTO MNEMONIC TABLE
F8C1: 4A                 LSR
F8C2: 4A        MNNDX2   LSR               ;1) 1XXX1010->00101XXX
F8C3: 09 20              ORA   #$20        ;2) XXXYYY01->00111XXX
F8C5: 88                 DEY               ;3) XXXYYY10->00110XXX
F8C6: D0 FA              BNE   MNNDX2      ;4) XXXYY100->00100XXX
```

```
F8C8: C8                  INY              ;5) XXXXX000->000XXXXX
F8C9: 88         MNNDX3   DEY
F8CA: D0 F2               BNE   MNNDX1
F8CC: 60                  RTS
F8CD: FF FF FF            DFB   $FF,$FF,$FF
F8D0: 20 82 F8   INSTDSP  JSR   INSDS1     ;GEN FMT, LEN BYTES
F8D3: 48                  PHA              ;SAVE MNEMONIC TABLE INDEX
F8D4: B1 3A      PRNTOP   LDA   (PCL),Y
F8D6: 20 DA FD            JSR   PRBYTE
F8D9: A2 01               LDX   #$01       ;PRINT 2 BLANKS
F8DB: 20 4A F9   PRNTBL   JSR   PRBL2
F8DE: C4 2F               CPY   LENGTH     ;PRINT INST (1-3 BYTES)
F8E0: C8                  INY              ;IN A 12 CHR FIELD
F8E1: 90 F1               BCC   PRNTOP
F8E3: A2 03               LDX   #$03       ;CHAR COUNT FOR MNEMONIC PRINT
F8E5: C0 04               CPY   #$04
F8E7: 90 F2               BCC   PRNTBL
F8E9: 68                  PLA              ;RECOVER MNEMONIC INDEX
F8EA: A8                  TAY
F8EB: B9 C0 F9            LDA   MNEML,Y
F8EE: 85 2C               STA   LMNEM      ;FETCH 3-CHAR MNEMONIC
F8F0: B9 00 FA            LDA   MNEMR,Y    ;  (PACKED IN 2-BYTES)
F8F3: 85 2D               STA   RMNEM
F8F5: A9 00      PRMN1    LDA   #$00
F8F7: A0 05               LDY   #$05
F8F9: 06 2D      PRMN2    ASL   RMNEM      ;SHIFT 5 BITS OF
F8FB: 26 2C               ROL   LMNEM      ;  CHARACTER INTO A
F8FD: 2A                  ROL              ;    (CLEARS CARRY)
F8FE: 88                  DEY
F8FF: D0 F8               BNE   PRMN2
F901: 69 BF               ADC   #$BF       ;ADD "?" OFFSET
F903: 20 ED FD            JSR   COUT       ;OUTPUT A CHAR OF MNEM
F906: CA                  DEX
F907: D0 EC               BNE   PRMN1
F909: 20 48 F9            JSR   PRBLNK     ;OUTPUT 3 BLANKS
F90C: A4 2F               LDY   LENGTH
F90E: A2 06               LDX   #$06       ;CNT FOR 6 FORMAT BITS
F910: E0 03      PRADR1   CPX   #$03
F912: F0 1C               BEQ   PRADR5     ;IF X=3 THEN ADDR.
F914: 06 2E      PRADR2   ASL   FORMAT
F916: 90 0E               BCC   PRADR3
F918: BD B3 F9            LDA   CHAR1-1,X
F91B: 20 ED FD            JSR   COUT
F91E: BD B9 F9            LDA   CHAR2-1,X
F921: F0 03               BEQ   PRADR3
F923: 20 ED FD            JSR   COUT
F926: CA         PRADR3   DEX
F927: D0 E7               BNE   PRADR1
F929: 60                  RTS
F92A: 88         PRADR4   DEY
F92B: 30 E7               BMI   PRADR2
F92D: 20 DA FD            JSR   PRBYTE
F930: A5 2E      PRADR5   LDA   FORMAT
F932: C9 E8               CMP   #$E8       ;HANDLE REL ADR MODE
F934: B1 3A               LDA   (PCL),Y    ;SPECIAL (PRINT TARGET,
F936: 90 F2     275           BCC   PRADR4     ;  NOT OFFSET)
F938: 20 56 F9  276  RELADR   JSR   PCADJ3
F93B: AA        277           TAX              ;PCL,PCH+OFFSET+1 TO A,Y
F93C: E8        278           INX
F93D: D0 01     279           BNE   PRNTYX     ;+1 TO Y,X
F93F: C8        280           INY
F940: 98        281  PRNTYX   TYA
F941: 20 DA FD  282  PRNTAX   JSR   PRBYTE     ;OUTPUT TARGET ADR
F944: 8A        283  PRNTX    TXA              ;  OF BRANCH AND RETURN
F945: 4C DA FD  284           JMP   PRBYTE
F948: A2 03     285  PRBLNK   LDX   #$03       ;BLANK COUNT
F94A: A9 A0     286  PRBL2    LDA   #$A0       ;LOAD A SPACE
F94C: 20 ED FD  287  PRBL3    JSR   COUT       ;OUTPUT A BLANK
F94F: CA        288           DEX
F950: D0 F8     289           BNE   PRBL2      ;LOOP UNTIL COUNT=0
F952: 60        290           RTS
F953: 38        291  PCADJ    SEC              ;0=1-BYTE, 1=2-BYTE
F954: A5 2F     292  PCADJ2   LDA   LENGTH     ;  2=3-BYTE
F956: A4 3B     293  PCADJ3   LDY   PCH
F958: AA        294           TAX              ;TEST DISPLACEMENT SIGN
F959: 10 01     295           BPL   PCADJ4     ;  (FOR REL BRANCH)
F95B: 88        296           DEY              ;EXTEND NEG BY DEC PCH
F95C: 65 3A     297  PCADJ4   ADC   PCL
F95E: 90 01     298           BCC   RTS2       ;PCL+LENGTH(OR DISPL)+1 TO A
F960: C8        299           INY              ;  CARRY INTO Y (PCH)
F961: 60        300  RTS2     RTS
                301  * FMT1 BYTES:    XXXXXXY0 INSTRS
                302  * IF Y=0         THEN LEFT HALF BYTE
                303  * IF Y=1         THEN RIGHT HALF BYTE
                304  *                   (X=INDEX)
F962: 04 20 54  305  FMT1     DFB   $04,$20,$54,$30,$0D
F965: 30 0D
F967: 80 04 90  306           DFB   $80,$04,$90,$03,$22
F96A: 03 22
F96C: 54 33 0D  307           DFB   $54,$33,$0D,$80,$04
F96F: 80 04
F971: 90 04 20  308           DFB   $90,$04,$20,$54,$33
F974: 54 33
F976: 0D 80 04  309           DFB   $0D,$80,$04,$90,$04
F979: 90 04
F97B: 20 54 3B  310           DFB   $20,$54,$3B,$0D,$80
F97E: 0D 80
F980: 04 90 00  311           DFB   $04,$90,$00,$22,$44
F983: 22 44
F985: 33 0D C8  312           DFB   $33,$0D,$C8,$44,$00
F988: 44 00
F98A: 11 22 44  313           DFB   $11,$22,$44,$33,$0D
F98D: 33 0D
F98F: C8 44 A9  314           DFB   $C8,$44,$A9,$01,$22
F992: 01 22
F994: 44 33 0D  315           DFB   $44,$33,$0D,$80,$04
F997: 80 04
F999: 90 01 22  316           DFB   $90,$01,$22,$44,$33
F99C: 44 33
F99E: 0D 80 04  317           DFB   $0D,$80,$04,$90
```

```
F9A1: 90
F9A2: 26 31 87  318           DFB   $26,$31,$87,$9A ;$ZZXXXY01 INSTR'S
F9A5: 9A
F9A6: 00        319  FMT2     DFB   $00         ;ERR
F9A7: 21        320           DFB   $21         ;IMM
F9A8: 81        321           DFB   $81         ;Z-PAGE
F9A9: 82        322           DFB   $82         ;ABS
F9AA: 00        323           DFB   $00         ;IMPLIED
F9AB: 00        324           DFB   $00         ;ACCUMULATOR
F9AC: 59        325           DFB   $59         ;(ZPAG,X)
F9AD: 4D        326           DFB   $4D         ;(ZPAG),Y
F9AE: 91        327           DFB   $91         ;ZPAG,X
F9AF: 92        328           DFB   $92         ;ABS,X
F9B0: 86        329           DFB   $86         ;ABS,Y
F9B1: 4A        330           DFB   $4A         ;(ABS)
F9B2: 85        331           DFB   $85         ;ZPAG,Y
F9B3: 9D        332           DFB   $9D         ;RELATIVE
F9B4: AC A9 AC  333  CHAR1    ASC   ",),#($"
F9B7: A3 A8 A4
F9BA: D9 00 D8  334  CHAR2    DFB   $D9,$00,$D8,$A4,$A4,$00
F9BD: A4 A4 00
                335  *CHAR2: "Y",0,"X$$",0
                336  * MNEML IS OF FORM:
                337  *  (A) XXXXX000
                338  *  (B) XXXYY100
                339  *  (C) 1XXX1010
                340  *  (D) XXXYYY10
                341  *  (E) XXXYYY01
                342  *      (X=INDEX)
F9C0: 1C 8A 1C  343  MNEML    DFB   $1C,$8A,$1C,$23,$5D,$8B
F9C3: 23 5D 8B
F9C6: 1B A1 9D  344           DFB   $1B,$A1,$9D,$8A,$1D,$23
F9C9: 8A 1D 23
F9CC: 9D 8B 1D  345           DFB   $9D,$8B,$1D,$A1,$00,$29
F9CF: A1 00 29
F9D2: 19 AE 69  346           DFB   $19,$AE,$69,$A8,$19,$23
F9D5: A8 19 23
F9D8: 24 53 1B  347           DFB   $24,$53,$1B,$23,$24,$53
F9DB: 23 24 53
F9DE: 19 A1     348           DFB   $19,$A1    ;(A) FORMAT ABOVE
F9E0: 00 1A 5B  349           DFB   $00,$1A,$5B,$5B,$A5,$69
F9E3: 5B A5 69
F9E6: 24 24     350           DFB   $24,$24    ;(B) FORMAT
F9E8: AE AE A8  351           DFB   $AE,$AE,$A8,$AD,$29,$00
F9EB: AD 29 00
F9EE: 7C 00     352           DFB   $7C,$00    ;(C) FORMAT
F9F0: 15 9C 6D  353           DFB   $15,$9C,$6D,$9C,$A5,$69
F9F3: 9C A5 69
F9F6: 29 53     354           DFB   $29,$53    ;(D) FORMAT
F9F8: 84 13 34  355           DFB   $84,$13,$34,$11,$A5,$69
F9FB: 11 A5 69
F9FE: 23 A0     356           DFB   $23,$A0    ;(E) FORMAT
FA00: D8 62 5A  357  MNEMR    DFB   $D8,$62,$5A,$48,$26,$62
FA03: 48 26 62
FA06: 94 88 54  358           DFB   $94,$88,$54,$44,$C8,$54
```

```
FA09: 44 C8 54
FA0C: 68 44 E8  359           DFB   $68,$44,$E8,$94,$00,$B4
FA0F: 94 00 B4
FA12: 08 84 74  360           DFB   $08,$84,$74,$B4,$28,$6E
FA15: B4 28 6E
FA18: 74 F4 CC  361           DFB   $74,$F4,$CC,$4A,$72,$F2
FA1B: 4A 72 F2
FA1E: A4 8A     362           DFB   $A4,$8A    ;(A) FORMAT
FA20: 00 AA A2  363           DFB   $00,$AA,$A2,$A2,$74,$74
FA23: A2 74 74
FA26: 74 72     364           DFB   $74,$72    ;(B) FORMAT
FA28: 44 68 B2  365           DFB   $44,$68,$B2,$32,$B2,$00
FA2B: 32 B2 00
FA2E: 22 00     366           DFB   $22,$00    ;(C) FORMAT
FA30: 1A 1A 26  367           DFB   $1A,$1A,$26,$26,$72,$72
FA33: 26 72 72
FA36: 88 C8     368           DFB   $88,$C8    ;(D) FORMAT
FA38: C4 CA 26  369           DFB   $C4,$CA,$26,$48,$44,$44
FA3B: 48 44 44
FA3E: A2 C8     370           DFB   $A2,$C8    ;(E) FORMAT
FA40: FF FF FF  371           DFB   $FF,$FF,$FF
FA43: 20 D0 F8  372 STEP      JSR   INSTDSP    ;DISASSEMBLE ONE INST
FA46: 68        373           PLA              ;  AT (PCL,H)
FA47: 85 2C     374           STA   RTNL       ;ADJUST TO USER
FA49: 68        375           PLA              ;  STACK. SAVE
FA4A: 85 2D     376           STA   RTNH       ;  RTN ADR.
FA4C: A2 08     377           LDX   #$08
FA4E: BD 10 FB  378 XQINIT    LDA   INITBL-1,X ;INIT XEQ AREA
FA51: 95 3C     379           STA   XQT,X
FA53: CA        380           DEX
FA54: D0 F8     381           BNE   XQINIT
FA56: A1 3A     382           LDA   (PCL,X)    ;USER OPCODE BYTE
FA58: F0 42     383           BEQ   XBRK       ;SPECIAL IF BREAK
FA5A: A4 2F     384           LDY   LENGTH     ;LEN FROM DISASSEMBLY
FA5C: C9 20     385           CMP   #$20
FA5E: F0 59     386           BEQ   XJSR       ;HANDLE JSR, RTS, JMP,
FA60: C9 60     387           CMP   #$60       ;  JMP (), RTI SPECIAL
FA62: F0 45     388           BEQ   XRTS
FA64: C9 4C     389           CMP   #$4C
FA66: F0 5C     390           BEQ   XJMP
FA68: C9 6C     391           CMP   #$6C
FA6A: F0 59     392           BEQ   XJMPAT
FA6C: C9 40     393           CMP   #$40
FA6E: F0 35     394           BEQ   XRTI
FA70: 29 1F     395           AND   #$1F
FA72: 49 14     396           EOR   #$14
FA74: C9 04     397           CMP   #$04       ;COPY USER INST TO XEQ AREA
FA76: F0 02     398           BEQ   XQ2        ;  WITH TRAILING NOPS
FA78: B1 3A     399 XQ1       LDA   (PCL),Y    ;CHANGE REL BRANCH
FA7A: 99 3C 00  400 XQ2       STA   XQT,Y      ;  DISP TO 4 FOR
FA7D: 88        401           DEY              ;  JMP TO BRANCH OR
FA7E: 10 F8     402           BPL   XQ1        ;  NBRANCH FROM XEQ.
FA80: 20 3F FF  403           JSR   RESTORE    ;RESTORE USER REG CONTENTS.
FA83: 4C 3C 00  404           JMP   XQT        ;XEQ USER OP FROM RAM
FA86: 85 45     405 IRQ       STA   ACC        ;  (RETURN TO NBRANCH)
```

```
FA88: 68        406            PLA
FA89: 48        407            PHA              ;**IRQ HANDLER
FA8A: 0A        408            ASL
FA8B: 0A        409            ASL
FA8C: 0A        410            ASL
FA8D: 30 03     411            BMI   BREAK      ;TEST FOR BREAK
FA8F: 6C FE 03  412            JMP   (IRQLOC)   ;USER ROUTINE VECTOR IN RAM
FA92: 28        413  BREAK     PLP
FA93: 20 4C FF  414            JSR   SAV1       ;SAVE REG'S ON BREAK
FA96: 68        415            PLA              ;  INCLUDING PC
FA97: 85 3A     416            STA   PCL
FA99: 68        417            PLA
FA9A: 85 3B     418            STA   PCH
FA9C: 20 82 F8  419  XBRK      JSR   INSDS1     ;PRINT USER PC.
FA9F: 20 DA FA  420            JSR   RGDSP1     ;  AND REG'S
FAA2: 4C 65 FF  421            JMP   MON        ;GO TO MONITOR
FAA5: 18        422  XRTI      CLC
FAA6: 68        423            PLA              ;SIMULATE RTI BY EXPECTING
FAA7: 85 48     424            STA   STATUS     ;  STATUS FROM STACK, THEN RTS
FAA9: 68        425  XRTS      PLA              ;RTS SIMULATION
FAAA: 85 3A     426            STA   PCL        ;  EXTRACT PC FROM STACK
FAAC: 68        427            PLA              ;  AND UPDATE PC BY 1 (LEN=0)
FAAD: 85 3B     428  PCINC2    STA   PCH
FAAF: A5 2F     429  PCINC3    LDA   LENGTH     ;UPDATE PC BY LEN
FAB1: 20 56 F9  430            JSR   PCADJ3
FAB4: 84 3B     431            STY   PCH
FAB6: 18        432            CLC
FAB7: 90 14     433            BCC   NEWPCL
FAB9: 18        434  XJSR      CLC
FABA: 20 54 F9  435            JSR   PCADJ2     ;UPDATE PC AND PUSH
FABD: AA        436            TAX              ;  ONTO STACH FOR
FABE: 98        437            TYA              ;  JSR SIMULATE
FABF: 48        438            PHA
FAC0: 8A        439            TXA
FAC1: 48        440            PHA
FAC2: A0 02     441            LDY   #$02
FAC4: 18        442  XJMP      CLC
FAC5: B1 3A     443  XJMPAT    LDA   (PCL),Y
FAC7: AA        444            TAX              ;LOAD PC FOR JMP,
FAC8: 88        445            DEY              ;  (JMP) SIMULATE.
FAC9: B1 3A     446            LDA   (PCL),Y
FACB: 86 3B     447            STX   PCH
FACD: 85 3A     448  NEWPCL    STA   PCL
FACF: B0 F3     449            BCS   XJMP
FAD1: A5 2D     450  RTNJMP    LDA   RTNH
FAD3: 48        451            PHA
FAD4: A5 2C     452            LDA   RTNL
FAD6: 48        453            PHA
FAD7: 20 8E FD  454  REGDSP    JSR   CROUT      ;DISPLAY USER REG
FADA: A9 45     455  RGDSP1    LDA   #ACC       ;  CONTENTS WITH
FADC: 85 40     456            STA   A3L        ;  LABELS
FADE: A9 00     457            LDA   #ACC/256
FAE0: 85 41     458            STA   A3H
FAE2: A2 FB     459            LDX   #$FB
FAE4: A9 A0     460  RDSP1     LDA   #$A0
```

```
FAE6: 20 ED FD           JSR   COUT
FAE9: BD 1E FA           LDA   RTBL-$FB,X
FAEC: 20 ED FD           JSR   COUT
FAEF: A9 BD              LDA   #$BD
FAF1: 20 ED FD           JSR   COUT
FAF4: B5 4A              LDA   ACC+5,X
FAF6: 20 DA FD           JSR   PRBYTE
FAF9: E8                 INX
FAFA: 30 E8              BMI   RDSP1
FAFC: 60                 RTS
FAFD: 18        BRANCH   CLC               ;BRANCH TAKEN,
FAFE: A0 01              LDY   #$01        ;  ADD LEN+2 TO PC
FB00: B1 3A              LDA   (PCL),Y
FB02: 20 56 F9           JSR   PCADJ3
FB05: 85 3A              STA   PCL
FB07: 98                 TYA
FB08: 38                 SEC
FB09: B0 A2              BCS   PCINC2
FB0B: 20 4A FF  NBRNCH   JSR   SAVE        ;NORMAL RETURN AFTER
FB0E: 38                 SEC               ;  XEQ USER OF
FB0F: B0 9E              BCS   PCINC3      ;GO UPDATE PC
FB11: EA        INITBL   NOP
FB12: EA                 NOP               ;DUMMY FILL FOR
FB13: 4C 0B FB           JMP   NBRNCH      ;  XEQ AREA
FB16: 4C FD FA           JMP   BRANCH
FB19: C1        RTBL     DFB   $C1
FB1A: D8                 DFB   $D8
FB1B: D9                 DFB   $D9
FB1C: D0                 DFB   $D0
FB1D: D3                 DFB   $D3
FB1E: AD 70 C0  PREAD    LDA   PTRIG       ;TRIGGER PADDLES
FB21: A0 00              LDY   #$00        ;INIT COUNT
FB23: EA                 NOP               ;COMPENSATE FOR 1ST COUNT
FB24: EA                 NOP
FB25: BD 64 C0  PREAD2   LDA   PADDL0,X    ;COUNT Y-REG EVERY
FB28: 10 04              BPL   RTS2D       ;  12 USEC
FB2A: C8                 INY
FB2B: D0 F8              BNE   PREAD2      ;  EXIT AT 255 MAX
FB2D: 88                 DEY
FB2E: 60        RTS2D    RTS
FB2F: A9 00     INIT     LDA   #$00        ;CLR STATUS FOR DEBUG
FB31: 85 48              STA   STATUS      ;  SOFTWARE
FB33: AD 56 C0           LDA   LORES
FB36: AD 54 C0           LDA   LOWSCR      ;INIT VIDEO MODE
FB39: AD 51 C0  SETTXT   LDA   TXTSET      ;SET FOR TEXT MODE
FB3C: A9 00              LDA   #$00        ;  FULL SCREEN WINDOW
FB3E: F0 0B              BEQ   SETWND
FB40: AD 50 C0  SETGR    LDA   TXTCLR      ;SET FOR GRAPHICS MODE
FB43: AD 53 C0           LDA   MIXSET      ;  LOWER 4 LINES AS
FB46: 20 36 F8           JSR   CLRTOP      ;  TEXT WINDOW
FB49: A9 14              LDA   #$14
FB4B: 85 22     SETWND   STA   WNDTOP      ;SET FOR 40 COL WINDOW
FB4D: A9 00              LDA   #$00        ;  TOP IN A-REG,
FB4F: 85 20              STA   WNDLFT      ;  BTTM AT LINE 24
FB51: A9 28              LDA   #$28
```

```
FB53: 85 21     516          STA   WNDWDTH
FB55: A9 18     517          LDA   #$18
FB57: 85 23     518          STA   WNDBTM     ;  VTAB TO ROW 23
FB59: A9 17     519          LDA   #$17
FB5B: 85 25     520  TABV    STA   CV         ;VTABS TO ROW IN A-REG
FB5D: 4C 22 FC  521          JMP   VTAB
FB60: 20 A4 FB  522  MULPM   JSR   MD1        ;ABS VAL OF AC AUX
FB63: A0 10     523  MUL     LDY   #$10       ;INDEX FOR 16 BITS
FB65: A5 50     524  MUL2    LDA   ACL        ;ACX * AUX + XTND
FB67: 4A        525          LSR              ; TO AC, XTND
FB68: 90 0C     526          BCC   MUL4       ;IF NO CARRY,
FB6A: 18        527          CLC              ; NO PARTIAL PROD.
FB6B: A2 FE     528          LDX   #$FE
FB6D: B5 54     529  MUL3    LDA   XTNDL+2,X  ;ADD MPLCND (AUX)
FB6F: 75 56     530          ADC   AUXL+2,X   ; TO PARTIAL PROD
FB71: 95 54     531          STA   XTNDL+2,X  ; (XTND)
FB73: E8        532          INX
FB74: D0 F7     533          BNE   MUL3
FB76: A2 03     534  MUL4    LDX   #$03
FB78: 76        535  MUL5    DFB   $76
FB79: 50        536          DFB   $50
FB7A: CA        537          DEX
FB7B: 10 FB     538          BPL   MUL5
FB7D: 88        539          DEY
FB7E: D0 E5     540          BNE   MUL2
FB80: 60        541          RTS
FB81: 20 A4 FB  542  DIVPM   JSR   MD1        ;ABS VAL OF AC, AUX.
FB84: A0 10     543  DIV     LDY   #$10       ;INDEX FOR 16 BITS
FB86: 06 50     544  DIV2    ASL   ACL
FB88: 26 51     545          ROL   ACH
FB8A: 26 52     546          ROL   XTNDL      ;XTND/AUX
FB8C: 26 53     547          ROL   XTNDH      ;  TO AC.
FB8E: 38        548          SEC
FB8F: A5 52     549          LDA   XTNDL
FB91: E5 54     550          SBC   AUXL       ;MOD TO XTND.
FB93: AA        551          TAX
FB94: A5 53     552          LDA   XTNDH
FB96: E5 55     553          SBC   AUXH
FB98: 90 06     554          BCC   DIV3
FB9A: 86 52     555          STX   XTNDL
FB9C: 85 53     556          STA   XTNDH
FB9E: E6 50     557          INC   ACL
FBA0: 88        558  DIV3    DEY
FBA1: D0 E3     559          BNE   DIV2
FBA3: 60        560          RTS
FBA4: A0 00     561  MD1     LDY   #$00       ;ABS VAL OF AC, AUX
FBA6: 84 2F     562          STY   SIGN       ;  WITH RESULT SIGN
FBA8: A2 54     563          LDX   #AUXL      ;  IN LSB OF SIGN.
FBAA: 20 AF FB  564          JSR   MD3
FBAD: A2 50     565          LDX   #ACL
FBAF: B5 01     566  MD3     LDA   LOC1,X     ;X SPECIFIES AC OR AUX
FBB1: 10 0D     567          BPL   MDRTS
FBB3: 38        568          SEC
FBB4: 98        569          TYA
FBB5: F5 00     570          SBC   LOC0,X     ;COMPL SPECIFIED REG
```

```
FBB7: 95 00     571            STA   LOC0,X       ;  IF NEG.
FBB9: 98        572            TYA
FBBA: F5 01     573            SBC   LOC1,X
FBBC: 95 01     574            STA   LOC1,X
FBBE: E6 2F     575            INC   SIGN
FBC0: 60        576  MDRTS     RTS
FBC1: 48        577  BASCALC   PHA                ;CALC BASE ADR IN BASL,H
FBC2: 4A        578            LSR                ;  FOR GIVEN LINE NO
FBC3: 29 03     579            AND   #$03         ;  0<=LINE NO.<=$17
FBC5: 09 04     580            ORA   #$04         ;ARG=000ABCDE, GENERATE
FBC7: 85 29     581            STA   BASH         ;  BASH=000001CD
FBC9: 68        582            PLA                ;  AND
FBCA: 29 18     583            AND   #$18         ;  BASL=EABAB000
FBCC: 90 02     584            BCC   BSCLC2
FBCE: 69 7F     585            ADC   #$7F
FBD0: 85 28     586  BSCLC2    STA   BASL
FBD2: 0A        587            ASL
FBD3: 0A        588            ASL
FBD4: 05 28     589            ORA   BASL
FBD6: 85 28     590            STA   BASL
FBD8: 60        591            RTS
FBD9: C9 87     592  BELL1     CMP   #$87         ;BELL CHAR? (CNTRL-G)
FBDB: D0 12     593            BNE   RTS2B        ;  NO, RETURN
FBDD: A9 40     594            LDA   #$40         ;DELAY .01 SECONDS
FBDF: 20 A8 FC  595            JSR   WAIT
FBE2: A0 C0     596            LDY   #$C0
FBE4: A9 0C     597  BELL2     LDA   #$0C         ;TOGGLE SPEAKER AT
FBE6: 20 A8 FC  598            JSR   WAIT         ;  1 KHZ FOR .1 SEC.
FBE9: AD 30 C0  599            LDA   SPKR
FBEC: 88        600            DEY
FBED: D0 F5     601            BNE   BELL2
FBEF: 60        602  RTS2B     RTS
FBF0: A4 24     603  STOADV    LDY   CH           ;CURSOR H INDEX TO Y-REG
FBF2: 91 28     604            STA   (BASL),Y     ;STORE CHAR IN LINE
FBF4: E6 24     605  ADVANCE   INC   CH           ;INCREMENT CURSOR H INDEX
FBF6: A5 24     606            LDA   CH           ;  (MOVE RIGHT)
FBF8: C5 21     607            CMP   WNDWDTH      ;BEYOND WINDOW WIDTH?
FBFA: B0 66     608            BCS   CR           ;  YES CR TO NEXT LINE
FBFC: 60        609  RTS3      RTS                ;  NO,RETURN
FBFD: C9 A0     610  VIDOUT    CMP   #$A0         ;CONTROL CHAR?
FBFF: B0 EF     611            BCS   STOADV       ;  NO,OUTPUT IT.
FC01: A8        612            TAY                ;INVERSE VIDEO?
FC02: 10 EC     613            BPL   STOADV       ;  YES, OUTPUT IT.
FC04: C9 8D     614            CMP   #$8D         ;CR?
FC06: F0 5A     615            BEQ   CR           ;  YES.
FC08: C9 8A     616            CMP   #$8A         ;LINE FEED?
FC0A: F0 5A     617            BEQ   LF           ;  IF SO, DO IT.
FC0C: C9 88     618            CMP   #$88         ;BACK SPACE? (CNTRL-H)
FC0E: D0 C9     619            BNE   BELL1        ;  NO, CHECK FOR BELL.
FC10: C6 24     620  BS        DEC   CH           ;DECREMENT CURSOR H INDEX
FC12: 10 E8     621            BPL   RTS3         ;IF POS, OK. ELSE MOVE UP
FC14: A5 21     622            LDA   WNDWDTH      ;SET CH TO WNDWDTH-1
FC16: 85 24     623            STA   CH
FC18: C6 24     624            DEC   CH           ;(RIGHTMOST SCREEN POS)
FC1A: A5 22     625  UP        LDA   WNDTOP       ;CURSOR V INDEX
```

```
FC1C: C5 25              CMP   CV
FC1E: B0 0B              BCS   RTS4       ;IF TOP LINE THEN RETURN
FC20: C6 25              DEC   CV         ;DEC CURSOR V-INDEX
FC22: A5 25     VTAB     LDA   CV         ;GET CURSOR V-INDEX
FC24: 20 C1 FB  VTABZ    JSR   BASCALC    ;GENERATE BASE ADR
FC27: 65 20              ADC   WNDLFT     ;ADD WINDOW LEFT INDEX
FC29: 85 28              STA   BASL       ;TO BASL
FC2B: 60        RTS4     RTS
FC2C: 49 C0     ESC1     EOR   #$C0       ;ESC?
FC2E: F0 28              BEQ   HOME       ;  IF SO, DO HOME AND CLEAR
FC30: 69 FD              ADC   #$FD       ;ESC-A OR B CHECK
FC32: 90 C0              BCC   ADVANCE    ;  A, ADVANCE
FC34: F0 DA              BEQ   BS         ;  B, BACKSPACE
FC36: 69 FD              ADC   #$FD       ;ESC-C OR D CHECK
FC38: 90 2C              BCC   LF         ;  C, DOWN
FC3A: F0 DE              BEQ   UP         ;  D, GO UP
FC3C: 69 FD              ADC   #$FD       ;ESC-E OR F CHECK
FC3E: 90 5C              BCC   CLREOL     ;  E, CLEAR TO END OF LINE
FC40: D0 E9              BNE   RTS4       ;  NOT F, RETURN
FC42: A4 24     CLREOP   LDY   CH         ;CURSOR H TO Y INDEX
FC44: A5 25              LDA   CV         ;CURSOR V TO A-REGISTER
FC46: 48        CLEOP1   PHA              ;SAVE CURRENT LINE ON STK
FC47: 20 24 FC           JSR   VTABZ      ;CALC BASE ADDRESS
FC4A: 20 9E FC           JSR   CLEOLZ     ;CLEAR TO EOL, SET CARRY
FC4D: A0 00              LDY   #$00       ;CLEAR FROM H INDEX=0 FOR REST
FC4F: 68                 PLA              ;INCREMENT CURRENT LINE
FC50: 69 00              ADC   #$00       ;(CARRY IS SET)
FC52: C5 23              CMP   WNDBTM     ;DONE TO BOTTOM OF WINDOW?
FC54: 90 F0              BCC   CLEOP1     ;  NO, KEEP CLEARING LINES
FC56: B0 CA              BCS   VTAB       ;  YES, TAB TO CURRENT LINE
FC58: A5 22     HOME     LDA   WNDTOP     ;INIT CURSOR V
FC5A: 85 25              STA   CV         ;  AND H-INDICES
FC5C: A0 00              LDY   #$00
FC5E: 84 24              STY   CH         ;THEN CLEAR TO END OF PAGE
FC60: F0 E4              BEQ   CLEOP1
FC62: A9 00     CR       LDA   #$00       ;CURSOR TO LEFT OF INDEX
FC64: 85 24              STA   CH         ;(RET CURSOR H=0)
FC66: E6 25     LF       INC   CV         ;INCR CURSOR V(DOWN 1 LINE)
FC68: A5 25              LDA   CV
FC6A: C5 23              CMP   WNDBTM     ;OFF SCREEN?
FC6C: 90 B6              BCC   VTABZ      ;  NO, SET BASE ADDR
FC6E: C6 25              DEC   CV         ;DECR CURSOR V (BACK TO BOTTOM)
FC70: A5 22     SCROLL   LDA   WNDTOP     ;START AT TOP OF SCRL WNDW
FC72: 48                 PHA
FC73: 20 24 FC           JSR   VTABZ      ;GENERATE BASE ADR
FC76: A5 28     SCRL1    LDA   BASL       ;COPY BASL,H
FC78: 85 2A              STA   BAS2L      ;  TO BAS2L,H
FC7A: A5 29              LDA   BASH
FC7C: 85 2B              STA   BAS2H
FC7E: A4 21              LDY   WNDWDTH    ;INIT Y TO RIGHTMOST INDEX
FC80: 88                 DEY              ;  OF SCROLLING WINDOW
FC81: 68                 PLA
FC82: 69 01              ADC   #$01       ;INCR LINE NUMBER
FC84: C5 23              CMP   WNDBTM     ;DONE?
FC86: B0 0D              BCS   SCRL3      ;  YES, FINISH
```

```
FC88: 48                 PHA
FC89: 20 24 FC           JSR   VTABZ        ;FORM BASL,H (BASE ADDR)
FC8C: B1 28    SCRL2     LDA   (BASL),Y     ;MOVE A CHR UP ON LINE
FC8E: 91 2A              STA   (BAS2L),Y
FC90: 88                 DEY                ;NEXT CHAR OF LINE
FC91: 10 F9              BPL   SCRL2
FC93: 30 E1              BMI   SCRL1        ;NEXT LINE (ALWAYS TAKEN)
FC95: A0 00    SCRL3     LDY   #$00         ;CLEAR BOTTOM LINE
FC97: 20 9E FC           JSR   CLEOLZ       ;GET BASE ADDR FOR BOTTOM LINE
FC9A: B0 86              BCS   VTAB         ;CARRY IS SET
FC9C: A4 24    CLREOL    LDY   CH           ;CURSOR H INDEX
FC9E: A9 A0    CLEOLZ    LDA   #$A0
FCA0: 91 28    CLEOL2    STA   (BASL),Y     ;STORE BLANKS FROM 'HERE'
FCA2: C8                 INY                ;  TO END OF LINES (WNDWDTH)
FCA3: C4 21              CPY   WNDWDTH
FCA5: 90 F9              BCC   CLEOL2
FCA7: 60                 RTS
FCA8: 38       WAIT      SEC
FCA9: 48       WAIT2     PHA
FCAA: E9 01    WAIT3     SBC   #$01
FCAC: D0 FC              BNE   WAIT3        ;1.0204 USEC
FCAE: 68                 PLA                ;(13+27/2*A+5/2*A*A)
FCAF: E9 01              SBC   #$01
FCB1: D0 F6              BNE   WAIT2
FCB3: 60                 RTS
FCB4: E6 42    NXTA4     INC   A4L          ;INCR 2-BYTE A4
FCB6: D0 02              BNE   NXTA1        ;  AND A1
FCB8: E6 43              INC   A4H
FCBA: A5 3C    NXTA1     LDA   A1L          ;INCR 2-BYTE A1.
FCBC: C5 3E              CMP   A2L
FCBE: A5 3D              LDA   A1H          ;  AND COMPARE TO A2
FCC0: E5 3F              SBC   A2H
FCC2: E6 3C              INC   A1L          ;  (CARRY SET IF >=)
FCC4: D0 02              BNE   RTS4B
FCC6: E6 3D              INC   A1H
FCC8: 60       RTS4B     RTS
FCC9: A0 4B    HEADR     LDY   #$4B         ;WRITE A*256 'LONG 1'
FCCB: 20 DB FC           JSR   ZERDLY       ;  HALF CYCLES
FCCE: D0 F9              BNE   HEADR        ;  (650 USEC EACH)
FCD0: 69 FE              ADC   #$FE
FCD2: B0 F5              BCS   HEADR        ;THEN A 'SHORT 0'
FCD4: A0 21              LDY   #$21         ;  (400 USEC)
FCD6: 20 DB FC WRBIT     JSR   ZERDLY       ;WRITE TWO HALF CYCLES
FCD9: C8                 INY                ;  OF 250 USEC ('0')
FCDA: C8                 INY                ;  OR 500 USEC ('0')
FCDB: 88       ZERDLY    DEY
FCDC: D0 FD              BNE   ZERDLY
FCDE: 90 05              BCC   WRTAPE       ;Y IS COUNT FOR
FCE0: A0 32              LDY   #$32         ;  TIMING LOOP
FCE2: 88       ONEDLY    DEY
FCE3: D0 FD              BNE   ONEDLY
FCE5: AC 20 C0 WRTAPE    LDY   TAPEOUT
FCE8: A0 2C              LDY   #$2C
FCEA: CA                 DEX
FCEB: 60                 RTS
```

```
FCEC: A2 08     RDBYTE   LDX   #$08        ;8 BITS TO READ
FCEE: 48        RDBYT2   PHA               ;READ TWO TRANSITIONS
FCEF: 20 FA FC           JSR   RD2BIT      ;  (FIND EDGE)
FCF2: 68                 PLA
FCF3: 2A                 ROL               ;NEXT BIT
FCF4: A0 3A              LDY   #$3A        ;COUNT FOR SAMPLES
FCF6: CA                 DEX
FCF7: D0 F5              BNE   RDBYT2
FCF9: 60                 RTS
FCFA: 20 FD FC  RD2BIT   JSR   RDBIT
FCFD: 88        RDBIT    DEY               ;DECR Y UNTIL
FCFE: AD 60 C0           LDA   TAPEIN      ; TAPE TRANSITION
FD01: 45 2F              EOR   LASTIN
FD03: 10 F8              BPL   RDBIT
FD05: 45 2F              EOR   LASTIN
FD07: 85 2F              STA   LASTIN
FD09: C0 80              CPY   #$80        ;SET CARRY ON Y
FD0B: 60                 RTS
FD0C: A4 24     RDKEY    LDY   CH
FD0E: B1 28              LDA   (BASL),Y    ;SET SCREEN TO FLASH
FD10: 48                 PHA
FD11: 29 3F              AND   #$3F
FD13: 09 40              ORA   #$40
FD15: 91 28              STA   (BASL),Y
FD17: 68                 PLA
FD18: 6C 38 00           JMP   (KSWL)      ;GO TO USER KEY-IN
FD1B: E6 4E     KEYIN    INC   RNDL
FD1D: D0 02              BNE   KEYIN2      ;INCR RND NUMBER
FD1F: E6 4F              INC   RNDH
FD21: 2C 00 C0  KEYIN2   BIT   KBD         ;KEY DOWN?
FD24: 10 F5              BPL   KEYIN       ;  LOOP
FD26: 91 28              STA   (BASL),Y    ;REPLACE FLASHING SCREEN
FD28: AD 00 C0           LDA   KBD         ;GET KEYCODE
FD2B: 2C 10 C0           BIT   KBDSTRB     ;CLR KEY STROBE
FD2E: 60                 RTS
FD2F: 20 0C FD  ESC      JSR   RDKEY       ;GET KEYCODE
FD32: 20 2C FC           JSR   ESC1        ;  HANDLE ESC FUNC.
FD35: 20 0C FD  RDCHAR   JSR   RDKEY       ;READ KEY
FD38: C9 9B              CMP   #$9B        ;ESC?
FD3A: F0 F3              BEQ   ESC         ;  YES, DON'T RETURN
FD3C: 60                 RTS
FD3D: A5 32     NOTCR    LDA   INVFLG
FD3F: 48                 PHA
FD40: A9 FF              LDA   #$FF
FD42: 85 32              STA   INVFLG      ;ECHO USER LINE
FD44: BD 00 02           LDA   IN,X        ;  NON INVERSE
FD47: 20 ED FD           JSR   COUT
FD4A: 68                 PLA
FD4B: 85 32              STA   INVFLG
FD4D: BD 00 02           LDA   IN,X
FD50: C9 88              CMP   #$88        ;CHECK FOR EDIT KEYS
FD52: F0 1D              BEQ   BCKSPC      ;  BS, CTRL-X
FD54: C9 98              CMP   #$98
FD56: F0 0A              BEQ   CANCEL
FD58: E0 F8              CPX   #$F8        ;MARGIN?
```

```
FD5A: 90 03              BCC   NOTCR1
FD5C: 20 3A FF           JSR   BELL        ;  YES, SOUND BELL
FD5F: E8        NOTCR1   INX               ;ADVANCE INPUT INDEX
FD60: D0 13              BNE   NXTCHAR
FD62: A9 DC     CANCEL   LDA   #$DC        ;BACKSLASH AFTER CANCELLED LINE
FD64: 20 ED FD           JSR   COUT
FD67: 20 8E FD  GETLNZ   JSR   CROUT       ;OUTPUT CR
FD6A: A5 33     GETLN    LDA   PROMPT
FD6C: 20 ED FD           JSR   COUT        ;OUTPUT PROMPT CHAR
FD6F: A2 01              LDX   #$01        ;INIT INPUT INDEX
FD71: 8A        BCKSPC   TXA               ;  WILL BACKSPACE TO 0
FD72: F0 F3              BEQ   GETLNZ
FD74: CA                 DEX
FD75: 20 35 FD  NXTCHAR  JSR   RDCHAR
FD78: C9 95              CMP   #PICK       ;USE SCREEN CHAR
FD7A: D0 02              BNE   CAPTST      ;  FOR CTRL-U
FD7C: B1 28              LDA   (BASL),Y
FD7E: C9 E0     CAPTST   CMP   #$E0
FD80: 90 02              BCC   ADDINP      ;CONVERT TO CAPS
FD82: 29 DF              AND   #$DF
FD84: 9D 00 02  ADDINP   STA   IN,X        ;ADD TO INPUT BUF
FD87: C9 8D              CMP   #$8D
FD89: D0 B2              BNE   NOTCR
FD8B: 20 9C FC           JSR   CLREOL      ;CLR TO EOL IF CR
FD8E: A9 8D     CROUT    LDA   #$8D
FD90: D0 5B              BNE   COUT
FD92: A4 3D     PRA1     LDY   A1H         ;PRINT CR,A1 IN HEX
FD94: A6 3C              LDX   A1L
FD96: 20 8E FD  PRYX2    JSR   CROUT
FD99: 20 40 F9           JSR   PRNTYX
FD9C: A0 00              LDY   #$00
FD9E: A9 AD              LDA   #$AD        ;PRINT '-'
FDA0: 4C ED FD           JMP   COUT
FDA3: A5 3C     XAM8     LDA   A1L
FDA5: 09 07              ORA   #$07        ;SET TO FINISH AT
FDA7: 85 3E              STA   A2L         ;  MOD 8=7
FDA9: A5 3D              LDA   A1H
FDAB: 85 3F              STA   A2H
FDAD: A5 3C     MODSCHK  LDA   A1L
FDAF: 29 07              AND   #$07
FDB1: D0 03              BNE   DATAOUT
FDB3: 20 92 FD  XAM      JSR   PRA1
FDB6: A9 A0     DATAOUT  LDA   #$A0
FDB8: 20 ED FD           JSR   COUT        ;OUTPUT BLANK
FDBB: B1 3C              LDA   (A1L),Y
FDBD: 20 DA FD           JSR   PRBYTE      ;OUTPUT BYTE IN HEX
FDC0: 20 BA FC           JSR   NXTA1
FDC3: 90 E8              BCC   MODSCHK     ;CHECK IF TIME TO,
FDC5: 60        RTS4C    RTS               ;  PRINT ADDR
FDC6: 4A        XAMPM    LSR               ;DETERMINE IF MON
FDC7: 90 EA              BCC   XAM         ;  MODE IS XAM
FDC9: 4A                 LSR               ;  ADD, OR SUB
FDCA: 4A                 LSR
FDCB: A5 3E              LDA   A2L
FDCD: 90 02              BCC   ADD
```

```
FDCF: 49 FF                EOR   #$FF       ;SUB: FORM 2'S COMPLEMENT
FDD1: 65 3C      ADD       ADC   A1L
FDD3: 48                   PHA
FDD4: A9 BD                LDA   #$BD
FDD6: 20 ED FD             JSR   COUT       ;PRINT '=', THEN RESULT
FDD9: 68                   PLA
FDDA: 48         PRBYTE    PHA              ;PRINT BYTE AS 2 HEX
FDDB: 4A                   LSR              ;  DIGITS, DESTROYS A-REG
FDDC: 4A                   LSR
FDDD: 4A                   LSR
FDDE: 4A                   LSR
FDDF: 20 E5 FD             JSR   PRHEXZ
FDE2: 68                   PLA
FDE3: 29 0F      PRHEX     AND   #$0F       ;PRINT HEX DIG IN A-REG
FDE5: 09 B0      PRHEXZ    ORA   #$B0       ;  LSB'S
FDE7: C9 BA                CMP   #$BA
FDE9: 90 02                BCC   COUT
FDEB: 69 06                ADC   #$06
FDED: 6C 36 00   COUT      JMP   (CSWL)     ;VECTOR TO USER OUTPUT ROUTINE
FDF0: C9 A0      COUT1     CMP   #$A0
FDF2: 90 02                BCC   COUTZ      ;DON'T OUTPUT CTRL'S INVERSE
FDF4: 25 32                AND   INVFLG     ;MASK WITH INVERSE FLAG
FDF6: 84 35      COUTZ     STY   YSAV1      ;SAV Y-REG
FDF8: 48                   PHA              ;SAV A-REG
FDF9: 20 FD FB             JSR   VIDOUT     ;OUTPUT A-REG AS ASCII
FDFC: 68                   PLA              ;RESTORE A-REG
FDFD: A4 35                LDY   YSAV1      ;  AND Y-REG
FDFF: 60                   RTS              ;  THEN RETURN
FE00: C6 34      BL1       DEC   YSAV
FE02: F0 9F                BEQ   XAM8
FE04: CA         BLANK     DEX              ;BLANK TO MON
FE05: D0 16                BNE   SETMDZ     ;AFTER BLANK
FE07: C9 BA                CMP   #$BA       ;DATA STORE MODE?
FE09: D0 BB                BNE   XAMPM      ;  NO, XAM, ADD, OR SUB
FE0B: 85 31      STOR      STA   MODE       ;KEEP IN STORE MODE
FE0D: A5 3E                LDA   A2L
FE0F: 91 40                STA   (A3L),Y    ;STORE AS LOW BYTE AS (A3)
FE11: E6 40                INC   A3L
FE13: D0 02                BNE   RTS5       ;INCR A3, RETURN
FE15: E6 41                INC   A3H
FE17: 60         RTS5      RTS
FE18: A4 34      SETMODE   LDY   YSAV       ;SAVE CONVERTED ':', '+',
FE1A: B9 FF 01             LDA   IN-1,Y     ;  '-', '.' AS MODE.
FE1D: 85 31      SETMDZ    STA   MODE
FE1F: 60                   RTS
FE20: A2 01      LT        LDX   #$01
FE22: B5 3E      LT2       LDA   A2L,X      ;COPY A2 (2 BYTES) TO
FE24: 95 42                STA   A4L,X      ;  A4 AND A5
FE26: 95 44                STA   A5L,X
FE28: CA                   DEX
FE29: 10 F7                BPL   LT2
FE2B: 60                   RTS
FE2C: B1 3C      MOVE      LDA   (A1L),Y    ;MOVE (A1 TO A2) TO
FE2E: 91 42                STA   (A4L),Y    ;  (A4)
FE30: 20 B4 FC             JSR   NXTA4
```

```
FE33: 90 F7              BCC   MOVE
FE35: 60                 RTS
FE36: B1 3C     VFY      LDA   (A1L),Y    ;VERIFY (A1 TO A2) WITH
FE38: D1 42              CMP   (A4L),Y    ;  (A4)
FE3A: F0 1C              BEQ   VFYOK
FE3C: 20 92 FD           JSR   PRA1
FE3F: B1 3C              LDA   (A1L),Y
FE41: 20 DA FD           JSR   PRBYTE
FE44: A9 A0              LDA   #$A0
FE46: 20 ED FD           JSR   COUT
FE49: A9 A8              LDA   #$A8
FE4B: 20 ED FD           JSR   COUT
FE4E: B1 42              LDA   (A4L),Y
FE50: 20 DA FD           JSR   PRBYTE
FE53: A9 A9              LDA   #$A9
FE55: 20 ED FD           JSR   COUT
FE58: 20 B4 FC  VFYOK    JSR   NXTA4
FE5B: 90 D9              BCC   VFY
FE5D: 60                 RTS
FE5E: 20 75 FE  LIST     JSR   A1PC       ;MOVE A1 (2 BYTES) TO
FE61: A9 14              LDA   #$14       ;  PC IF SPEC'D AND
FE63: 48        LIST2    PHA              ;  DISEMBLE 20 INSTRS
FE64: 20 D0 F8           JSR   INSTDSP
FE67: 20 53 F9           JSR   PCADJ      ;ADJUST PC EACH INSTR
FE6A: 85 3A              STA   PCL
FE6C: 84 3B              STY   PCH
FE6E: 68                 PLA
FE6F: 38                 SEC
FE70: E9 01              SBC   #$01       ;NEXT OF 20 INSTRS
FE72: D0 EF              BNE   LIST2
FE74: 60                 RTS
FE75: 8A        A1PC     TXA              ;IF USER SPEC'D ADR
FE76: F0 07              BEQ   A1PCRTS    ;  COPY FROM A1 TO PC
FE78: B5 3C     A1PCLP   LDA   A1L,X
FE7A: 95 3A              STA   PCL,X
FE7C: CA                 DEX
FE7D: 10 F9              BPL   A1PCLP
FE7F: 60        A1PCRTS  RTS
FE80: A0 3F     SETINV   LDY   #$3F       ;SET FOR INVERSE VID
FE82: D0 02              BNE   SETIFLG    ; VIA COUT1
FE84: A0 FF     SETNORM  LDY   #$FF       ;SET FOR NORMAL VID
FE86: 84 32     SETIFLG  STY   INVFLG
FE88: 60                 RTS
FE89: A9 00     SETKBD   LDA   #$00       ;SIMULATE PORT #0 INPUT
FE8B: 85 3E     INPORT   STA   A2L        ;  SPECIFIED (KEYIN ROUTINE)
FE8D: A2 38     INPRT    LDX   #KSWL
FE8F: A0 1B              LDY   #KEYIN
FE91: D0 08              BNE   IOPRT
FE93: A9 00     SETVID   LDA   #$00       ;SIMULATE PORT #0 OUTPUT
FE95: 85 3E     OUTPORT  STA   A2L        ;  SPECIFIED (COUT1 ROUTINE)
FE97: A2 36     OUTPRT   LDX   #CSWL
FE99: A0 F0              LDY   #COUT1
FE9B: A5 3E     IOPRT    LDA   A2L        ;SET RAM IN/OUT VECTORS
FE9D: 29 0F              AND   #$0F
FE9F: F0 06              BEQ   IOPRT1
```

```
FEA1: 09 C0     956          ORA   #IOADR/256
FEA3: A0 00     957          LDY   #$00
FEA5: F0 02     958          BEQ   IOPRT2
FEA7: A9 FD     959  IOPRT1  LDA   #COUT1/256
FEA9: 94 00     960  IOPRT2  STY   LOC0,X
FEAB: 95 01     961          STA   LOC1,X
FEAD: 60        962          RTS
FEAE: EA        963          NOP
FEAF: EA        964          NOP
FEB0: 4C 00 E0  965  XBASIC  JMP   BASIC      ;TO BASIC WITH SCRATCH
FEB3: 4C 03 E0  966  BASCONT JMP   BASIC2     ;CONTINUE BASIC
FEB6: 20 75 FE  967  GO      JSR   A1PC       ;ADR TO PC IF SPEC'D
FEB9: 20 3F FF  968          JSR   RESTORE    ;RESTORE META REGS
FEBC: 6C 3A 00  969          JMP   (PCL)      ;GO TO USER SUBR
FEBF: 4C D7 FA  970  REGZ    JMP   REGDSP     ;TO REG DISPLAY
FEC2: C6 34     971  TRACE   DEC   YSAV
FEC4: 20 75 FE  972  STEPZ   JSR   A1PC       ;ADR TO PC IF SPEC'D
FEC7: 4C 43 FA  973          JMP   STEP       ;TAKE ONE STEP
FECA: 4C F8 03  974  USR     JMP   USRADR     ;TO USR SUBR AT USRADR
FECD: A9 40     975  WRITE   LDA   #$40
FECF: 20 C9 FC  976          JSR   HEADR      ;WRITE 10-SEC HEADER
FED2: A0 27     977          LDY   #$27
FED4: A2 00     978  WR1     LDX   #$00
FED6: 41 3C     979          EOR   (A1L,X)
FED8: 48        980          PHA
FED9: A1 3C     981          LDA   (A1L,X)
FEDB: 20 ED FE  982          JSR   WRBYTE
FEDE: 20 BA FC  983          JSR   NXTA1
FEE1: A0 1D     984          LDY   #$1D
FEE3: 68        985          PLA
FEE4: 90 EE     986          BCC   WR1
FEE6: A0 22     987          LDY   #$22
FEE8: 20 ED FE  988          JSR   WRBYTE
FEEB: F0 4D     989          BEQ   BELL
FEED: A2 10     990  WRBYTE  LDX   #$10
FEEF: 0A        991  WRBYT2  ASL
FEF0: 20 D6 FC  992          JSR   WRBIT
FEF3: D0 FA     993          BNE   WRBYT2
FEF5: 60        994          RTS
FEF6: 20 00 FE  995  CRMON   JSR   BL1        ;HANDLE A CR AS BLANK
FEF9: 68        996          PLA              ;  THEN POP STACK
FEFA: 68        997          PLA              ;  AND RTN TO MON
FEFB: D0 6C     998          BNE   MONZ
FEFD: 20 FA FC  999  READ    JSR   RD2BIT     ;FIND TAPEIN EDGE
FF00: A9 16     1000         LDA   #$16
FF02: 20 C9 FC  1001         JSR   HEADR      ;DELAY 3.5 SECONDS
FF05: 85 2E     1002         STA   CHKSUM     ;INIT CHKSUM=$FF
FF07: 20 FA FC  1003         JSR   RD2BIT     ;FIND TAPEIN EDGE
FF0A: A0 24     1004 RD2     LDY   #$24       ;LOOK FOR SYNC BIT
FF0C: 20 FD FC  1005         JSR   RDBIT      ;  (SHORT 0)
FF0F: B0 F9     1006         BCS   RD2        ;  LOOP UNTIL FOUND
FF11: 20 FD FC  1007         JSR   RDBIT      ;SKIP SECOND SYNC H-CYCLE
FF14: A0 3B     1008         LDY   #$3B       ;INDEX FOR 0/1 TEST
FF16: 20 EC FC  1009 RD3     JSR   RDBYTE     ;READ A BYTE
FF19: 81 3C     1010         STA   (A1L,X)    ;STORE AT (A1)
```

```
FF1B: 45 2E     1011          EOR   CHKSUM
FF1D: 85 2E     1012          STA   CHKSUM     ;UPDATE RUNNING CHKSUM
FF1F: 20 BA FC  1013          JSR   NXTA1      ;INC A1, COMPARE TO A2
FF22: A0 35     1014          LDY   #$35       ;COMPENSATE 0/1 INDEX
FF24: 90 F0     1015          BCC   RD3        ;LOOP UNTIL DONE
FF26: 20 EC FC  1016          JSR   RDBYTE     ;READ CHKSUM BYTE
FF29: C5 2E     1017          CMP   CHKSUM
FF2B: F0 0D     1018          BEQ   BELL       ;GOOD, SOUND BELL AND RETURN
FF2D: A9 C5     1019 PRERR    LDA   #$C5
FF2F: 20 ED FD  1020          JSR   COUT       ;PRINT "ERR", THEN BELL
FF32: A9 D2     1021          LDA   #$D2
FF34: 20 ED FD  1022          JSR   COUT
FF37: 20 ED FD  1023          JSR   COUT
FF3A: A9 87     1024 BELL     LDA   #$87       ;OUTPUT BELL AND RETURN
FF3C: 4C ED FD  1025          JMP   COUT
FF3F: A5 48     1026 RESTORE  LDA   STATUS     ;RESTORE 6502 REG CONTENTS
FF41: 48        1027          PHA              ;  USED BY DEBUG SOFTWARE
FF42: A5 45     1028          LDA   ACC
FF44: A6 46     1029 RESTR1   LDX   XREG
FF46: A4 47     1030          LDY   YREG
FF48: 28        1031          PLP
FF49: 60        1032          RTS
FF4A: 85 45     1033 SAVE     STA   ACC        ;SAVE 6502 REG CONTENTS
FF4C: 86 46     1034 SAV1     STX   XREG
FF4E: 84 47     1035          STY   YREG
FF50: 08        1036          PHP
FF51: 68        1037          PLA
FF52: 85 48     1038          STA   STATUS
FF54: BA        1039          TSX
FF55: 86 49     1040          STX   SPNT
FF57: D8        1041          CLD
FF58: 60        1042          RTS
FF59: 20 84 FE  1043 RESET    JSR   SETNORM    ;SET SCREEN MODE
FF5C: 20 2F FB  1044          JSR   INIT       ;  AND INIT KBD/SCREEN
FF5F: 20 93 FE  1045          JSR   SETVID     ;  AS I/O DEV'S
FF62: 20 89 FE  1046          JSR   SETKBD
FF65: D8        1047 MON      CLD              ;MUST SET HEX MODE!
FF66: 20 3A FF  1048          JSR   BELL
FF69: A9 AA     1049 MONZ     LDA   #$AA       ;'*' PROMPT FOR MON
FF6B: 85 33     1050          STA   PROMPT
FF6D: 20 67 FD  1051          JSR   GETLNZ     ;READ A LINE
FF70: 20 C7 FF  1052          JSR   ZMODE      ;CLEAR MON MODE, SCAN IDX
FF73: 20 A7 FF  1053 NXTITM   JSR   GETNUM     ;GET ITEM, NON-HEX
FF76: 84 34     1054          STY   YSAV       ;  CHAR IN A-REG
FF78: A0 17     1055          LDY   #$17       ;  X-REG=0 IF NO HEX INPUT
FF7A: 88        1056 CHRSRCH  DEY
FF7B: 30 E8     1057          BMI   MON        ;NOT FOUND, GO TO MON
FF7D: D9 CC FF  1058          CMP   CHRTBL,Y   ;FIND CMND CHAR IN TEL
FF80: D0 F8     1059          BNE   CHRSRCH
FF82: 20 BE FF  1060          JSR   TOSUB      ;FOUND, CALL CORRESPONDING
FF85: A4 34     1061          LDY   YSAV       ;  SUBROUTINE
FF87: 4C 73 FF  1062          JMP   NXTITM
FF8A: A2 03     1063 DIG      LDX   #$03
FF8C: 0A        1064          ASL
FF8D: 0A        1065          ASL              ;GOT HEX DIG,
FF8E: 0A                 ASL              ;  SHIFT INTO A2
FF8F: 0A                 ASL
FF90: 0A        NXTBIT   ASL
FF91: 26 3E              ROL   A2L
FF93: 26 3F              ROL   A2H
FF95: CA                 DEX              ;LEAVE X=$FF IF DIG
FF96: 10 F8              BPL   NXTBIT
FF98: A5 31     NXTBAS   LDA   MODE
FF9A: D0 06              BNE   NXTBS2     ;IF MODE IS ZERO
FF9C: B5 3F              LDA   A2H,X      ; THEN COPY A2 TO
FF9E: 95 3D              STA   A1H,X      ; A1 AND A3
FFA0: 95 41              STA   A3H,X
FFA2: E8        NXTBS2   INX
FFA3: F0 F3              BEQ   NXTBAS
FFA5: D0 06              BNE   NXTCHR
FFA7: A2 00     GETNUM   LDX   #$00       ;CLEAR A2
FFA9: 86 3E              STX   A2L
FFAB: 86 3F              STX   A2H
FFAD: B9 00 02  NXTCHR   LDA   IN,Y       ;GET CHAR
FFB0: C8                 INY
FFB1: 49 B0              EOR   #$B0
FFB3: C9 0A              CMP   #$0A
FFB5: 90 D3              BCC   DIG        ;IF HEX DIG, THEN
FFB7: 69 88              ADC   #$88
FFB9: C9 FA              CMP   #$FA
FFBB: B0 CD              BCS   DIG
FFBD: 60                 RTS
FFBE: A9 FE     TOSUB    LDA   #GO/256    ;PUSH HIGH-ORDER
FFC0: 48                 PHA              ;  SUBR ADR ON STK
FFC1: B9 E3 FF           LDA   SUBTBL,Y   ;PUSH LOW-ORDER
FFC4: 48                 PHA              ;  SUBR ADR ON STK
FFC5: A5 31              LDA   MODE
FFC7: A0 00     ZMODE    LDY   #$00       ;CLR MODE, OLD MODE
FFC9: 84 31              STY   MODE       ;  TO A-REG
FFCB: 60                 RTS              ; GO TO SUBR VIA RTS
FFCC: BC        CHRTBL   DFB   $BC        ;F("CTRL-C")
FFCD: B2                 DFB   $B2        ;F("CTRL-Y")
FFCE: BE                 DFB   $BE        ;F("CTRL-E")
FFCF: ED                 DFB   $ED        ;F("T")
FFD0: EF                 DFB   $EF        ;F("V")
FFD1: C4                 DFB   $C4        ;F("CTRL-K")
FFD2: EC                 DFB   $EC        ;F("S")
FFD3: A9                 DFB   $A9        ;F("CTRL-P")
FFD4: BB                 DFB   $BB        ;F("CTRL-B")
FFD5: A6                 DFB   $A6        ;F("-")
FFD6: A4                 DFB   $A4        ;F("+")
FFD7: 06                 DFB   $06        ;F("M") (F=EX-OR $B0+$89)
FFD8: 95                 DFB   $95        ;F("<")
FFD9: 07                 DFB   $07        ;F("N")
FFDA: 02                 DFB   $02        ;F("I")
FFDB: 05                 DFB   $05        ;F("L")
FFDC: F0                 DFB   $F0        ;F("W")
FFDD: 00                 DFB   $00        ;F("G")
FFDE: EB                 DFB   $EB        ;F("R")
FFDF: 93                 DFB   $93        ;F(":")
```

```
FFE0: A7        1121            DFB   $A7         ;F(".")
FFE1: C6        1122            DFB   $C6         ;F("CR")
FFE2: 99        1123            DFB   $99         ;F(BLANK)
FFE3: B2        1124 SUBTBL     DFB   BASCONT-1
FFE4: C9        1125            DFB   USR-1
FFE5: BE        1126            DFB   REGZ-1
FFE6: C1        1127            DFB   TRACE-1
FFE7: 35        1128            DFB   VFY-1
FFE8: 8C        1129            DFB   INPRT-1
FFE9: C3        1130            DFB   STEPZ-1
FFEA: 96        1131            DFB   OUTPRT-1
FFEB: AF        1132            DFB   XBASIC-1
FFEC: 17        1133            DFB   SETMODE-1
FFED: 17        1134            DFB   SETMODE-1
FFEE: 2B        1135            DFB   MOVE-1
FFEF: 1F        1136            DFB   LT-1
FFF0: 83        1137            DFB   SETNORM-1
FFF1: 7F        1138            DFB   SETINV-1
FFF2: 5D        1139            DFB   LIST-1
FFF3: CC        1140            DFB   WRITE-1
FFF4: B5        1141            DFB   GO-1
FFF5: FC        1142            DFB   READ-1
FFF6: 17        1143            DFB   SETMODE-1
FFF7: 17        1144            DFB   SETMODE-1
FFF8: F5        1145            DFB   CRMON-1
FFF9: 03        1146            DFB   BLANK-1
FFFA: FB        1147            DFB   NMI         ;NMI VECTOR
FFFB: 03        1148            DFB   NMI/256
FFFC: 59        1149            DFB   RESET       ;RESET VECTOR
FFFD: FF        1150            DFB   RESET/256
FFFE: 86        1151            DFB   IRQ         ;IRQ VECTOR
FFFF: FA        1152            DFB   IRQ/256
                1153 XQTNZ      EQU   $3C
```

###