```
                        ;29-MAY-80      NEWMVH
                                .TITLE "MICRO FLICK - J. HUETHER"
0028                    SCNEND  =40     SCREEN END PLUS 1
0050                    RITEND=80
00A0                    PRTEND=160
0050                    PBTEND=80
                        ; CURFRM OFFSETS
0003                    BAKGND=3
0004                    P0=4
0005                    P1=5
0006                    P2=6
0007                    P3=7
0008                    X0=8
0009                    X1=9
000A                    X2=10
000B                    X3=11
000C                    Y0=12
000D                    Y1=13
000E                    Y2=14
000F                    Y3=15
                        ; PLAYER DATA OFFSETS
0004                    DTSTRT=4
0003                    DATLEN=3
                        ; LIST ELEMENT LENGTHS
0010                    FELELN=16
0002                    OELELN=2
0002                    BELELN  =2
0008                    MAXBAC  =8
0008                    FLDLEN  =8
0008                    MAXMSG  =8
0400                    GXPM=$400
                        ;
                        ;
0080                    Z       =$80
                                *=GXPM+$180
0580        3800        GXMISL  *=*+$80
0600                    GXP0    *=*+$80
0680                    GXP1    *=*+$80
0700                    GXP2    *=*+$80
0780                    GXP3    *=*+$80
0800                            V=*
                        ;
                        ;
                                *=$BFFA
BFFA 01 A0                      .WORD   CSTART
BFFC 00 04                      .BYTE   0,4
BFFE 00 A0                      .WORD   INIT
                                *=Z
0080                    DTAB=*
0080                    S1L     *=*+2
0082                    S1H     *=*+2
0084                    S2L     *=*+2
0086                    S2H     *=*+2
0088                    MEMA    *=*+2
008A                    MEMB    *=*+2
008C                    MSP     *=*+2
008E                    MDP     *=*+2
0090                    MBC     *=*+2
0092                    ACC     *=*+2
0094                    LS1L    *=*+2
```

3000 - 44BC

```
0096                    LS1H    *=*+2
0098                    LS2L    *=*+2
009A                    LS2H    *=*+2
009C                    DS1L    *=*+2
009E                    DS1H    *=*+2
00A0                    DS2L    *=*+2
00A2                    DS2H    *=*+2
00A4                    PUPDAT  *=*+2
00A6                    SCNPTR  *=*+2
00A8                    BLNKPT  *=*+2
00AA                    CHRPTR  *=*+2
00AC                    RAMPTR  *=*+2
00AE                    DLPTR   *=*+2
00B0                    PZJUMP  *=*+4
00B4                    PTR     *=*+2
00B6                    CURFRM  *=*+FELELN
00C6                    PRVFRM  *=*+FELELN
00D6                            Z=*
                                *=V
0800                    RTCL    *=*+1
0801                    CURSTT  *=*+1
0802                    PRVSTT  *=*+1
0803                    NXTSTT  *=*+1
0804                    CHGFLG  *=*+1
0805                    MSPEED  *=*+1
0806                    SAVKEY  *=*+1
0807                    CLSLCT  *=*+1
0808                    HWID    *=*+1
0809                    VWID    *=*+1
080A                    PTOP    *=*+1
080B                    PBOT    *=*+1
080C                    FRMCTR  *=*+1
080D                    OBJCTR  *=*+1
080E                    BACCTR  *=*+1
080F                    XPOS    *=*+1
0810                    YPOS    *=*+1
0811                    HORPO   *=*+4
0815                    HORMO   *=*+4
0819                    OLDVPO  *=*+4
081D                    OLDVMO  *=*+4
0821                    VPOSPO  *=*+4
0825                    VPOSMO  *=*+4
0829                    HEIGHT  *=*+4
082D                    OLDHGT  *=*+4
0831                    BITOFF  *=*+4
0835                    CURPLR  *=*+1
0836                    MODENO  *=*+1
0837                    PLBND   *=*+1
0838                    PRBND   *=*+1
0839                    MODE    *=*+1
083A                    INKCLR  *=*+1
083B                    HDELTA  *=*+8
0843                    VDELTA  *=*+8
084B                    VBFLAG  *=*+1
084C                    FBYTWD  *=*+1
084D                    BYTSTR  *=*+1
084E                    PIXWID  *=*+1
084F                    PIXLEN  *=*+1
0850                    SOFFND  *=*+1
0851                    BYTEND  *=*+1
```

```
0852                SOFFST  *=*+1
0853                BYTWID  *=*+1
0854                LEFT    *=*+1
0855                TOP     *=*+1
0856                RIGHT   *=*+1
0857                BOTTOM  *=*+1
0858                LB      *=*+1
0859                TB      *=*+1
085A                RB      *=*+1
085B                BB      *=*+1
085C                ENDX    *=*+1
085D                ENDY    *=*+1
085E                XBOX    *=*+2
0860                YBOX    *=*+2
0862                PTINDX  *=*+1
0863                UPDFLG  *=*+1
0864                MODFLG  *=*+1
0865                PRVTRO  *=*+1
0866                COLOUR  *=*+1
0867                FILVAL  *=*+1
0868                TMP1    *=*+1
0869                TMP2    *=*+1
086A                        V=*
```

```
                        ;
                        ; INITIALIZATION
                        ;
                                *=$A000
A000 60                 INIT    RTS
                        ;
                        ;
                        ; ASSUME O.S. CLEARS RAM ???????????????
                        ;
A001 A0 F8              CSTART  LDY     #VBHND
A003 A2 AC                      LDX     #VBHND/256
A005 A9 07                      LDA     #7      VVBLKD
A007 20 5C E4                   JSR     SETVBV
                        ; SET UP PAGE ZERO INDIRECT JSR
A00A A9 20                      LDA     #$20
A00C 85 B0                      STA     PZJUMP
A00E A9 60                      LDA     #$60
A010 85 B3                      STA     PZJUMP+3
                        ; OPEN IOCB2 TO MODE 5
A012 A2 20                      LDX     #$20    IOCB2
A014 A9 03                      LDA     #OPEN
A016 9D 42 03                   STA     ICCOM,X
A019 A9 10                      LDA     #MXDMOD
A01B 9D 4A 03                   STA     ICAX1,X
A01E A9 05                      LDA     #5      GRAFIX MODE 5 - 1K
A020 9D 4B 03                   STA     ICAX2,X
A023 A9 B1                      LDA     #SDSPLY
A025 9D 44 03                   STA     ICBAL,X
A028 A9 A2                      LDA     #SDSPLY/256
A02A 9D 45 03                   STA     ICBAH,X
A02D 20 56 E4                   JSR     CIOV
                        ; SET UP DLIST AND CHAR POINTERS
A030 AD 30 02                   LDA     SDLSTL
A033 85 AE                      STA     DLPTR
A035 AD 31 02                   LDA     SDLSTH
A038 85 AF                      STA     DLPTR+1
A03A A0 2E                      LDY     #46     BYTE OFFSET IN DLIST FOR CHAR. ADDRESS
A03C B1 AE                      LDA     (DLPTR),Y
A03E 85 AA                      STA     CHRPTR
A040 C8                         INY
A041 B1 AE                      LDA     (DLPTR),Y
A043 85 AB                      STA     CHRPTR+1
                        ; SET RAM AND BLANK SCREEN POINTERS
A045 A5 58                      LDA     SAVMSC
A047 85 AC                      STA     RAMPTR
A049 85 A8                      STA     BLNKPT
A04B A5 59                      LDA     SAVMSC+1
A04D 85 AD                      STA     RAMPTR+1
A04F 85 A9                      STA     BLNKPT+1
                        ; CLEAR CURRENT SCREEN (VIA RAMPTR)
A051 A9 00                      LDA     #0
A053 8D 66 08                   STA     COLOUR
A056 20 C5 B0                   JSR     FILSCN
                        ; SET UP DATA STRUCTURE PARAMETERS INITIALLY
A059 A9 00                      LDA     #0
A05B 8D 41 0C                   STA     MOVLEN
A05E 8D 0D 08                   STA     OBJCTR
A061 8D 40 0C                   STA     OLSTLN
A064 8D 0E 08                   STA     BACCTR
A067 8D 3F 0C                   STA     BLSTLN
```

```
A06A A9 FF                  LDA     #$FF
A06C 8D 0C 08               STA     FRMCTR
                    ; CLEAR CURFRM AND PRVFRM
A06F A0 0F                  LDY     #FELELN-1
A071 A9 FF                  LDA     #$FF
A073 99 B6 00       CFRMLP  STA     CURFRM,Y
A076 99 C6 00               STA     PRVFRM,Y
A079 88                     DEY
A07A 10 F7                  BPL     CFRMLP
                    ; CLEAR PLAYER AREAS

A07C A9 00                  LDA     #0
A07E 8D 67 08               STA     FILVAL
A081 A9 80                  LDA     #$80
A083 85 90                  STA     MBC
A085 A9 02                  LDA     #2
A087 85 91                  STA     MBC+1
A089 A9 80                  LDA     #GXMISL
A08B 85 8E                  STA     MDP
A08D A9 05                  LDA     #GXMISL/256
A08F 85 8F                  STA     MDP+1
A091 20 7B AD               JSR     FIL
                    ;
                    ;
                    ;
A094 A9 2E                  LDA     #$2E    STND PF,MSSL DMA ENABL,INST FTCH ENABL
A096 8D 2F 02               STA     SDMCTL
A099 A9 03                  LDA     #3
A09B 8D 1D D0               STA     GRACTL
A09E A9 00                  LDA     #0
A0A0 8D 01 D4               STA     CHACTL
A0A3 A9 04                  LDA     #GXPM/256
A0A5 8D 07 D4               STA     PMBASE
A0A8 A9 E0                  LDA     #CHRORG/256     CHARACTER TABLE ADDRESS
A0AA 8D F4 02               STA     CHBAS
                    ; SET MODE FOR NOW!!!!!!!!!!!!!!!!!!!
A0AD A9 01                  LDA     #1      PLYR,PF,BAK
A0AF 8D 6F 02               STA     GPRIOR
A0B2 A9 C0                  LDA     #$C0    ENABL NMI VBLNK INT,AND DLIST INT
A0B4 8D 0E D4               STA     NMIEN
A0B7 A5 10                  LDA     POKMSK
A0B9 09 40                  ORA     #$40
A0BB 85 10                  STA     POKMSK
A0BD 8D 0E D2               STA     IRGEN
                    ; OPEN IOCB1 TO KEYBOARD DEVICE
A0C0 A2 10                  LDX     #$10    IOCB1
A0C2 A9 03                  LDA     #OPEN
A0C4 9D 42 03               STA     ICCOM,X
A0C7 A9 AE                  LDA     #KEYBRD
A0C9 9D 44 03               STA     ICBAL,X
A0CC A9 A2                  LDA     #KEYBRD/256
A0CE 9D 45 03               STA     ICBAH,X
A0D1 A9 04                  LDA     #OPNIN  INPUT ONLY
A0D3 9D 4A 03               STA     ICAX1,X
A0D6 20 56 E4               JSR     CIOV
                    ;
                    ; SET UP MEM ALLOCATION POINTERS FOR 2 AREAS
                    ;
                    ; SET UP MEMLO
                    ;
```

```
A0D9 A9 2F              LDA     #BACLST
A0DB 85 86              STA     S2H
A0DD A9 0C              LDA     #BACLST/256
A0DF 85 87              STA     S2H+1
A0E1 A9 6F              LDA     #FRMLST
A0E3 8D E7 02           STA     MEMLO
A0E6 85 80              STA     S1L
A0E8 A9 08              LDA     #FRMLST/256
A0EA 8D E8 02           STA     MEMLO+1
A0ED 85 81              STA     S1L+1
A0EF 20 2F B3           JSR     MINIT
A0F2 20 0A AF           JSR     S1TLS1
                ;
A0F5 A9 42              LDA     #BACLST+19
A0F7 85 80              STA     S1L
A0F9 A9 0C              LDA     #BACLST+19/256
A0FB 85 81              STA     S1L+1
A0FD AD E5 02           LDA     MEMTOP
A100 85 86              STA     S2H
A102 AD E6 02           LDA     MEMTOP+1
A105 85 87              STA     S2H+1
A107 20 2F B3           JSR     MINIT   INITS S1H AND S2L - USE REGION 2
A10A 20 22 AF           JSR     S1TDS1
                ;
                ; INITIALIZE POKEY
                ;
A10D A9 03              LDA     #3
A10F 8D 32 02           STA     SSKCTL
A112 8D 0F D2           STA     SKCTL
A115 A9 00              LDA     #0
A117 8D 08 D2           STA     AUDCTL
                ; SET COLORS
A11A A0 03              LDY     #3
A11C B9 F4 AC   CLRLOD  LDA     COLRTB,Y
A11F 99 C0 02           STA     PCOLR0,Y
A122 88                 DEY
A123 10 F7              BPL     CLRLOD
                ; SET PLAYER SIZE TO NORMAL
A125 A2 03              LDX     #3
A127 A9 00              LDA     #0
A129 9D 08 D0   SIZLP   STA     SIZEP0,X
A12C CA                 DEX
A12D 10 FA              BPL     SIZLP
A12F A9 00              LDA     #0
A131 8D 0C D0           STA     SIZEM
                ;
A134 A9 09              LDA     #9
A136 8D 05 08           STA     MSPEED
A139 A9 03              LDA     #3
A13B 8D 3A 08           STA     INKCLR
                ; INSERT PF CURSOR WRITE HERE???????????
                ; SET UP PLAYER AND MISSILE SPEEDS
A13E A2 07              LDX     #7
A140 A9 01              LDA     #1
A142 9D 43 08   SPDLP   STA     VDELTA,X
A145 9D 3B 08           STA     HDELTA,X
A148 CA                 DEX
A149 10 F7              BPL     SPDLP
                ; SET POINTER TO BLANK BACKGROUND INTO BACLST
A14B A0 FF              LDY     #$FF
```

```
A14D 20 A2 AD                JSR     BACACC
A150 A0 01                   LDY     #1
A152 B9 A8 00      LOOP99    LDA     BLNKPT,Y
A155 91 B4                   STA     (PTR),Y
A157 88                      DEY
A158 10 F8                   BPL     LOOP99
                   ;
A15A 20 64 AF                JSR     CSCNBD
                   ; CLEAR OLD CURSOR
A15D AE 1D 08                LDX     OLDVMO
A160 20 C5 B2                JSR     ERSMSL
                   ; POSITION NEW CURSOR --- CANNED FOR NOW!!!!!!!!!!!!
A163 A2 28                   LDX     #40
A165 8E 25 08                STX     VPOSMO
A168 8E 1D 08                STX     OLDVMO
A16B A9 50                   LDA     #80
A16D 8D 15 08                STA     HORMO
                   ; WRITE NEW CURSOR
A170 20 B6 B2                JSR     WRTMSL
A173 A9 1A                   LDA     #26       INIT STATE
A175 8D 01 08                STA     CURSTT
                   ; BUILD SCREEN
                   ;
A178 20 77 AF                JSR     BLDSCN
A17B 20 9D AE      MAIN      JSR     WFVBFG
                   ;
                   ; MOVE CURSOR
A17E 20 D2 B1                JSR     CURSOR
                   ;
A181 20 33 A3                JSR     RDKEY
A184 D0 F5                   BNE     MAIN
A186 AD 01 08      CSTATE    LDA     CURSTT
A189 8D 02 08                STA     PRVSTT
A18C AD 03 08                LDA     NXTSTT
A18F 8D 01 08                STA     CURSTT
A192 0A                      ASL     A
A193 AA                      TAX
A194 BD B1 A1                LDA     JMPTBL,X
A197 85 B1                   STA     PZJUMP+1
A199 BD B2 A1                LDA     JMPTBL+1,X
A19C 85 B2                   STA     PZJUMP+2
A19E 20 B0 00                JSR     PZJUMP
A1A1 F0 E3                   BEQ     CSTATE  NO ERRORS
A1A3 E0 09         ERRCHK    CPX     #MAXMSG+1
A1A5 B0 DF                   BCS     CSTATE
A1A7 CA                      DEX
A1A8 8A                      TXA                ERROR #
A1A9 0A                      ASL     A
A1AA A8                      TAY
                   ; SELECT ERROR MESSAGE
A1AB 20 CC AE                JSR     MESAGE
A1AE 4C 7B A1                JMP     MAIN
                   ;
                   ;
A1B1 8F A5 85      JMPTBL    .WORD   ADDFRM,MAKBAK,CHGFRM,MAIN,ERSCRN
A1B4 A4 DA A5
A1B7 7B A1 76
A1BA AC
A1BB AE A9 7B                .WORD   CASSRD,MAIN,MAIN,MAIN,MAIN
A1BE A1 7B A1
```

```
A1C1 7B A1 7B
A1C4 A1
A1C5 2B AB 6C                .WORD   DELBAK,LOCPLR,MODPLR,MAIN,MAIN
A1C8 A6 03 A8
A1CB 7B A1 7B
A1CE A1
A1CF 8A A3 7B                .WORD   MAKPLR,MAIN,RUN,FRMSEQ,CASSAV
A1D2 A1 AD A7
A1D5 09 A6 AD
A1D8 A8
A1D9 7B A1 F5                .WORD   MAIN,BACSEQ,SETSPD,MAIN,MAIN
A1DC AB AC AC
A1DF 7B A1 7B
A1E2 A1
A1E3 97 AA                   .WORD   ZERMEM
                             ;
A1E5 F5 A1 14        MSGTBL  .WORD   MSG1,MSG2,MSG3,MSG4,MSG5,MSG6,MSG7,MSG8
A1E8 A2 26 A2
A1EB 38 A2 4C
A1EE A2 5E A2
A1F1 7C A2 96
A1F4 A2
                             ;
A1F5 4E 4F 20        MSG1    .BYTE   "NO OBJECTS AVAIL IN THIS FRAME",$9B
A1F8 4F 42 4A
A1FB 45 43 54
A1FE 53 20 41
A201 56 41 49
A204 4C 20 49
A207 4E 20 54
A20A 48 49 53
A20D 20 46 52
A210 41 4D 45
A213 9B
A214 4E 4F 20        MSG2    .BYTE   "NO OBJECT TO SAVE",$9B
A217 4F 42 4A
A21A 45 43 54
A21D 20 54 4F
A220 20 53 41
A223 56 45 9B
A226 4E 4F 54        MSG3    .BYTE   "NOT ENOUGH MEMORY",$9B
A229 20 45 4E
A22C 4F 55 47
A22F 48 20 4D
A232 45 4D 4F
A235 52 59 9B
A238 4F 42 4A        MSG4    .BYTE   "OBJECT NOT IN FRAME",$9B
A23B 45 43 54
A23E 20 4E 4F
A241 54 20 49
A244 4E 20 46
A247 52 41 4D
A24A 45 9B
A24C 4F 55 54        MSG5    .BYTE   "OUT OF LIST RANGE",$9B
A24F 20 4F 46
A252 20 4C 49
A255 53 54 20
A258 52 41 4E
A25B 47 45 9B
A25E 43 41 4E        MSG6    .BYTE   "CAN'T DELETE BLANK BACKGROUND",$9B
```

```
A261 27 54 20
A264 44 45 4C
A267 45 54 45
A26A 20 42 4C
A26D 41 4E 4B
A270 20 42 41
A273 43 4B 47
A276 52 4F 55
A279 4E 44 9B
A27C 49 4C 4C        MSG7    .BYTE   "ILLEGAL FUNCTION TRANSFER",$9B
A27F 45 47 41
A282 4C 20 46
A285 55 4E 43
A288 54 49 4F
A28B 4E 20 54
A28E 52 41 4E
A291 53 46 45
A294 52 9B
A296 49 4C 4C        MSG8    .BYTE   "ILLEGAL SUBFUNCTION KEY",$9B
A299 45 47 41
A29C 4C 20 53
A29F 55 42 46
A2A2 55 4E 43
A2A5 54 49 4F
A2A8 4E 20 4B
A2AB 45 59 9B

A2AE 4B 3A 9B        KEYBRD  .BYTE   "K:",$9B
A2B1 53 3A 9B        SDSPLY  .BYTE   "S:",$9B
A2B4 43 3A 9B        CASS    .BYTE   "C:",$9B
```

```
A2B7 4A 1B FE      STATBL  .BYTE   $4A,$1B,$FE,0   A
A2BA 00
A2BB AA 1B BA              .BYTE   $AA,$1B,$8A,0   B
A2BE 00
A2BF 4A 1B FE              .BYTE   $4A,$1B,$FE,0   C
A2C2 00
A2C3 08 02 26              .BYTE   $8,$2,$26,0     D
A2C6 00
A2C7 E2 03 60              .BYTE   $E2,$3,$60,0    E
A2CA 00
A2CB 08 42 76              .BYTE   $8,$42,$76,0    F
A2CE 00
A2CF FC FF FF              .BYTE   $FC,$FF,$FF,$80 G
A2D2 80
A2D3 4A 42 76              .BYTE   $4A,$42,$76,$80 H
A2D6 80
A2D7 E8 1B 8C              .BYTE   $E8,$1B,$8C,0   I
A2DA 00
A2DB EE 3B FE              .BYTE   $EE,$3B,$FE,$80 J
A2DE 80
A2DF 0A 42 74              .BYTE   $A,$42,$74,0    K
A2E2 00
A2E3 EA 0B 8C              .BYTE   $EA,$B,$8C,0    L
A2E6 00
A2E7 EA 13 8C              .BYTE   $EA,$13,$8C,0   M
A2EA 00
A2EB EA 1B 8C              .BYTE   $EA,$1B,$8C,0   N
A2EE 00
A2EF 12 80 00              .BYTE   $12,$80,$0,0    O
A2F2 00
A2F3 EA 1B 8C              .BYTE   $EA,$1B,$8C,0   P
A2F6 00
A2F7 EA 1B 0C              .BYTE   $EA,$1B,$C,0    Q
A2FA 00
A2FB 01 00 00              .BYTE   $1,0,0,0        R
A2FE 00
A2FF 4A 1B DC              .BYTE   $4A,$1B,$DC,0   S
A302 00
A303 4A 5B FE              .BYTE   $4A,$5B,$FE,$40 T
A306 40
A307 EA 1B 84              .BYTE   $EA,$1B,$84,0   U
A30A 00
A30B 0A 26 38              .BYTE   $A,$26,$38,0    V
A30E 00
A30F 4A 42 34              .BYTE   $4A,$42,$34,$80 W
A312 80
A313 0A 02 74              .BYTE   $A,$2,$74,0     X
A316 00
A317 4A 43 22              .BYTE   $4A,$43,$22,$40 Y
A31A 40
A31B 4C 43 02              .BYTE   $4C,$43,$2,$80  Z
A31E 80
A31F 44 41 02              .BYTE   $44,$41,$2,$80  INIT
A322 80
```

```
                        ; ECHO KEY AND DO RETURN
A323 A2 00      ECHO    LDX     #0      EDITOR
A325 AC 06 08           LDY     SAVKEY
A328 20 5F B0           JSR     PUT1CH
A32B A2 00              LDX     #0
A32D A0 9B              LDY     #$9B
A32F 20 5F B0           JSR     PUT1CH
A332 60                 RTS
                        ; SUBROUTINE RDKEY
A333 AD FC 02   RDKEY   LDA     CH
A336 C9 FF              CMP     #$FF
A338 D0 03              BNE     *+5
A33A A0 01              LDY     #1
A33C 60                 RTS
A33D 20 71 B0           JSR     GETKEE
A340 8D 06 08           STA     SAVKEY
A343 C9 41              CMP     #'A
A345 90 40              BCC     NOFKEY
A347 C9 5B              CMP     #$5B
A349 B0 3C              BCS     NOFKEY
A34B AD 01 08   FCNKEY  LDA     CURSTT  0-26
A34E 0A                 ASL     A
A34F 0A                 ASL     A
A350 A8                 TAY
A351 AD 06 08           LDA     SAVKEY
A354 38                 SEC
A355 E9 41              SBC     #'A     4(0-26)
A357 8D 03 08           STA     NXTSTT
A35A 4A                 LSR     A
A35B 4A                 LSR     A
A35C 4A                 LSR     A
A35D 8D 68 08           STA     TMP1
A360 98                 TYA
A361 18                 CLC
A362 6D 68 08           ADC     TMP1
A365 A8                 TAY
A366 AD 03 08           LDA     NXTSTT
A369 29 07              AND     #7
A36B 49 07              EOR     #7
A36D AA                 TAX
A36E B9 B7 A2           LDA     STATBL,Y
A371 3D E4 AC           AND     ONETBL,X
A374 F0 09              BEQ     NOALOW
A376 AD 06 08           LDA     SAVKEY
A379 20 23 A3           JSR     ECHO
A37C A0 00              LDY     #0
A37E 60                 RTS
A37F A0 0C      NOALOW  LDY     #12     2 X 6
A381 20 CC AE           JSR     MESAGE
A384 A0 03              LDY     #3
A386 60                 RTS
A387 A0 02      NOFKEY  LDY     #2
A389 60                 RTS
                ;
                ;
```

```
                        ; MAKPLR
                        ;
                        ; CHECK FOR AVAILABLE PLAYER ENTRY IN CURFRM!!!!!!!!!!
                        ;
                        ;
                        ; SEARCH CURFRM FOR AVAILABLE PLAYER???????
A38A 20 77 AF   MAKPLR  JSR     BLDSCN
A38D A9 01              LDA     #1
A38F 8D 07 08           STA     CLSLCT
                        ;
                        ;
A392 20 9D AE   MAKPLP  JSR     WFVBFG
A395 20 D2 B1           JSR     CURSOR
A398 20 33 A3           JSR     RDKEY
A39B D0 03              BNE     *+5
A39D 4C 82 A4           JMP     ENDPLR
A3A0 C0 02              CPY     #2
A3A2 F0 22              BEQ     SUBKYP
A3A4 4C 92 A3           JMP     MAKPLP
A3A7 20 9D AE   NOKEY   JSR     WFVBFG
A3AA 20 D2 B1           JSR     CURSOR
A3AD 20 33 A3           JSR     RDKEY
A3B0 F0 60              BEQ     EXITP
A3B2 C0 02              CPY     #2
A3B4 F0 2E              BEQ     SKEYP
                        ; TEST FOR BUTTON PUSHED
A3B6 A2 00              LDX     #0      STICK NO.
A3B8 BD 84 02           LDA     STRIG0,X
A3BB D0 EA              BNE     NOKEY
                        ;
                        ;
                        ; SET UP PLAYER INDEX X FOR MODE
A3BD AE 35 08           LDX     CURPLR
A3C0 20 76 AE           JSR     INKPOS
                        ;
A3C3 4C A7 A3   M123    JMP     NOKEY
A3C6 AD 06 08   SUBKYP  LDA     SAVKEY
A3C9 C9 20              CMP     #'      SPACE KEY
A3CB D0 3D              BNE     ISKEYP
A3CD A2 03              LDX     #3
A3CF B5 BA      TAVAIL  LDA     CURFRM+PO,X
A3D1 C9 FF              CMP     #$FF
A3D3 F0 06              BEQ     CNTMAK
A3D5 CA                 DEX
A3D6 10 F7              BPL     TAVAIL
A3D8 A2 01              LDX     #1      NO PLAYERS AVAIL IN THIS FRAME
A3DA 60                 RTS
                        ;
                        ;
                        ; SET PLAYER BOUNDS FOR MODE
A3DB 8E 35 08   CNTMAK  STX     CURPLR
A3DE 20 A8 AE           JSR     CPLRBD
A3E1 4C A7 A3           JMP     NOKEY
A3E4 AD 06 08   SKEYP   LDA     SAVKEY
A3E7 C9 30      PCHK0   CMP     #'0
A3E9 D0 05              BNE     P111
A3EB A9 00              LDA     #0
A3ED 4C F6 A3           JMP     P222
A3F0 C9 31      P111    CMP     #'1
A3F2 D0 0E              BNE     ILSKYP
```

```
A3F4 A9 01              LDA     #1
A3F6 8D 07 08   P222    STA     CLSLCT
A3F9 AD 06 08           LDA     SAVKEY
A3FC 20 23 A3           JSR     ECHO
A3FF 4C A7 A3           JMP     NOKEY
A402 A0 0E      ILSKYP  LDY     #14
A404 20 CC AE           JSR     MESAGE
A407 4C A7 A3           JMP     NOKEY
A40A A0 0E      ISKEYP  LDY     #14
A40C 20 CC AE           JSR     MESAGE
A40F 4C 92 A3           JMP     MAKPLP
A412 AE 35 08   EXITP   LDX     CURPLR
A415 20 14 AE           JSR     VRTSIZ
A418 F0 01              BEQ     *+3
A41A 60                 RTS
                ; DO DIFFERENT STUFF HERE DEPENDING ON MODE
                ; GET PDATA MEMORY
A41B 20 16 AF           JSR     DS1TS1
A41E AE 35 08           LDX     CURPLR
A421 BD 29 08           LDA     HEIGHT,X
A424 18                 CLC
A425 69 04              ADC     #4
A427 85 8A              STA     MEMB
A429 A9 00              LDA     #0
A428 85 8B              STA     MEMB+1
A42D A5 82              LDA     S1H
A42F 85 88              STA     MEMA
A431 A5 83              LDA     S1H+1
A433 85 89              STA     MEMA+1
A435 20 3E B3           JSR     MALLOC
A438 F0 03              BEQ     *+5
A43A A2 03              LDX     #3
A43C 60                 RTS
                ; RESTORE DS1 POINTER
                ; DO MEM AVAIL MESSAGE HERE!!!!!!!!!!!!
A43D 20 22 AF           JSR     S1TDS1
                ; SAVE PDATA ADDRESS TO PTR
A440 A5 88              LDA     MEMA
A442 85 B4              STA     PTR
A444 A5 89              LDA     MEMA+1
A446 85 B5              STA     PTR+1
                ; WRITE DATA
A448 AE 35 08           LDX     CURPLR
A44B 20 D9 AD           JSR     SAVPLR
                ; SAVE PLAYER COORDS TO CURFRM
A44E AD 40 0C           LDA     OLSTLN
A451 20 C9 AD           JSR     SAVCRD
                ; GET OBJLST MEMORY
A454 20 FE AE           JSR     LS1TS1
A457 A9 02              LDA     #OELELN
A459 85 8A              STA     MEMB
A45B A9 00              LDA     #0
A45D 85 8B              STA     MEMB+1
A45F A5 84              LDA     S2L
A461 85 88              STA     MEMA
A463 A5 85              LDA     S2L+1
A465 85 89              STA     MEMA+1
A467 20 3E B3           JSR     MALLOC
A46A F0 03              BEQ     *+5
A46C A2 03              LDX     #3
```

```
A46E 60                 RTS
                ; WRITE OBJLST PTR
                ; DO MEM AVAIL MESSAGE HERE!!!!!!!!!
A46F 20 0A AF           JSR     S1TLS1
A472 EE 40 0C           INC     OLSTLN
A475 A0 01              LDY     #1
A477 B9 B4 00   LOOP9B  LDA     PTR,Y
A47A 91 88              STA     (MEMA),Y
A47C 88                 DEY
A47D 10 F8              BPL     LOOP9B
                ; RE-ESTABLISH CURSOR BOUNDS TO SCREEN BOUNDS
A47F 20 64 AF           JSR     CSCNBD
A482 A2 00      ENDPLR  LDX     #0
A484 60                 RTS
                ;
```

```
                        ; MAKBAK
                        ; PLOP OUT CURRENT BACKGROUND ONLY
A485 20 77 AF       MAKBAK  JSR     BLDSCN
                        ; SET BOUNDS FOR CURSOR FOR SCREEN
A488 20 64 AF               JSR     CSCNBD
                        ;
                        ; GETMEM???????????????
                        ; GET MEMORY FOR NEW BACKGROUND
A48B 20 16 AF               JSR     DS1TS1
A48E A9 22                  LDA     #$22    802 BYTES
A490 85 8A                  STA     MEMB
A492 A9 03                  LDA     #3
A494 85 8B                  STA     MEMB+1
                        ; 4K TEST
A496 A5 84                  LDA     S2L
A498 38                     SEC
A499 E5 8A                  SBC     MEMB
A49B 8D 68 08               STA     TMP1
A49E A5 85                  LDA     S2L+1
A4A0 E5 8B                  SBC     MEMB+1
A4A2 8D 69 08               STA     TMP2
A4A5 45 85                  EOR     S2L+1
A4A7 29 10                  AND     #$10
A4A9 F0 0B                  BEQ     ALBAC
A4AB AD 69 08               LDA     TMP2    4K BOUND
A4AE 09 0F                  ORA     #$0F
A4B0 85 85                  STA     S2L+1
A4B2 A9 FF                  LDA     #$FF
A4B4 85 84                  STA     S2L
A4B6 A5 84          ALBAC   LDA     S2L
A4B8 85 88                  STA     MEMA
A4BA A5 85                  LDA     S2L+1
A4BC 85 89                  STA     MEMA+1
A4BE 20 3E B3               JSR     MALLOC
A4C1 F0 03                  BEQ     *+5
A4C3 A2 03                  LDX     #3
A4C5 60                     RTS
                        ; DO MEM AVAIL MESSAGE HERE!!!!!!!!!!!
A4C6 20 22 AF               JSR     S1TDS1
                        ; WRITE DATA
A4C9 A9 20                  LDA     #$20    800 BYTES
A4CB 85 90                  STA     MBC
A4CD A9 03                  LDA     #$3
A4CF 85 91                  STA     MBC+1
                        ; GET BAK CURRENTLY SHOWING AND MOVE TO NEWLY ALLOCATED AREA
A4D1 A0 04                  LDY     #4
A4D3 B1 AE                  LDA     (DLPTR),Y
A4D5 85 8C                  STA     MSP
A4D7 C8                     INY
A4D8 B1 AE                  LDA     (DLPTR),Y
A4DA 85 8D                  STA     MSP+1
A4DC A5 89                  LDA     MEMA+1
A4DE 85 8F                  STA     MDP+1
A4E0 85 AD                  STA     RAMPTR+1
A4E2 A5 88                  LDA     MEMA
A4E4 18                     CLC
A4E5 69 02                  ADC     #2      BACKGROUND DATA START
A4E7 85 8E                  STA     MDP
A4E9 85 AC                  STA     RAMPTR
A4EB 90 04                  BCC     *+6
```

```
A4ED E6 BF                  INC     MDP+1
A4EF E6 AD                  INC     RAMPTR+1
A4F1 20 5F B4               JSR     MOVIA
                    ; SHOW NEW GBACKGROUND AREA
A4F4 A0 04                  LDY     #4
A4F6 A5 AC                  LDA     RAMPTR
A4F8 91 AE                  STA     (DLPTR),Y
A4FA C8                     INY
A4FB A5 AD                  LDA     RAMPTR+1
A4FD 91 AE                  STA     (DLPTR),Y
                    ;
A4FF 20 9D AE       MAKBLP  JSR     WFVBFG
A502 20 33 A3               JSR     RDKEY
A505 F0 5E                  BEQ     LVMBAK
A507 C0 02                  CPY     #2
A509 F0 22                  BEQ     SUBKYB
A50B B0 00                  BCS     MPFCRS
                    ;
A50D 20 D2 B1       MPFCRS  JSR     CURSOR
A510 A2 00                  LDX     #0
A512 BD 84 02               LDA     STRIG0,X
A515 F0 03                  BEQ     INKPF
A517 4C FF A4               JMP     MAKBLP
                    ; INK PF
A51A AD 25 08       INKPF   LDA     VPOSM0
A51D 4A                     LSR     A
A51E A8                     TAY
A51F AD 15 08               LDA     HORM0
A522 4A                     LSR     A
A523 AA                     TAX
A524 AD 3A 08               LDA     INKCLR
A527 20 87 B1               JSR     WRITPX
A52A 4C FF A4               JMP     MAKBLP
A52D AD 06 08       SUBKYB  LDA     SAVKEY
A530 C9 30          CHKPF0  CMP     #'0
A532 D0 05                  BNE     CHKPF1
A534 A9 00                  LDA     #0
A536 4C 51 A5               JMP     SPFCLR
A539 C9 31          CHKPF1  CMP     #'1
A53B D0 05                  BNE     CHKPF2
A53D A9 01                  LDA     #1
A53F 4C 51 A5               JMP     SPFCLR
A542 C9 32          CHKPF2  CMP     #'2
A544 D0 05                  BNE     CHKPF3
A546 A9 02                  LDA     #2
A548 4C 51 A5               JMP     SPFCLR
A54B C9 33          CHKPF3  CMP     #'3
A54D D0 0E                  BNE     ILSKYB
A54F A9 03                  LDA     #3
A551 8D 3A 08       SPFCLR  STA     INKCLR
A554 AD 06 08               LDA     SAVKEY
A557 20 23 A3               JSR     ECHO
A55A 4C 0D A5               JMP     MPFCRS
A55D A0 0E          ILSKYB  LDY     #14
A55F 20 CC AE               JSR     MESAGE
A562 4C 0D A5               JMP     MPFCRS
                    ; SAVE OBJCTR TO CURFRM FOR BACKGROUND AND
                    ; INCREMENT OBJCTR AND SET OBJPTR
A565 AD 3F 0C       LVMBAK  LDA     BLSTLN
A568 A0 03                  LDY     #BAKGND
```

```
A56A 99 B6 00               STA     CURFRM,Y
A56D A8                     TAY
A56E 20 A2 AD               JSR     BACACC
                    ; TEST FOR MAXBAC???????????????
A571 EE 3F 0C               INC     BLSTLN
                    ; SAVE PTR TO BACLST
A574 A0 00                  LDY     #0
A576 A5 AC                  LDA     RAMPTR
A578 38                     SEC
A579 E9 02                  SBC     #2
A57B 8D 68 08               STA     TMP1
A57E A5 AD                  LDA     RAMPTR+1
A580 E9 00                  SBC     #0
A582 A0 01                  LDY     #1
A584 91 B4                  STA     (PTR),Y
A586 88                     DEY
A587 AD 68 08               LDA     TMP1
A58A 91 B4                  STA     (PTR),Y
A58C A2 00                  LDX     #0
A58E 60                     RTS
```

```
                        ; ADDFRM
                        ;
                        ; ALLOCATE FRAME MEMORY
A58F 20 77 AF       ADDFRM  JSR     BLDSCN
A592 20 FE AE               JSR     LS1TS1
A595 A9 10                  LDA     #FELELN
A597 85 8A                  STA     MEMB
A599 A9 00                  LDA     #0
A59B 85 8B                  STA     MEMB+1
                        ;
A59D EE 0C 08               INC     FRMCTR
A5A0 AC 0C 08               LDY     FRMCTR
A5A3 20 49 AF               JSR     FRMACC
A5A6 A5 B4                  LDA     PTR
A5A8 85 88                  STA     MEMA
A5AA A5 B5                  LDA     PTR+1
A5AC 85 89                  STA     MEMA+1
A5AE 20 3E B3               JSR     MALLOC
A5B1 F0 03                  BEQ     *+5
A5B3 A2 03                  LDX     #3
A5B5 60                     RTS
                        ; DO MEM AVAIL MESSAGE HERE!!!!!!!!!
A5B6 20 0A AF               JSR     S1TLS1
                        ; TRANSFER FROM CURFRM TO FRMLST
                        ; VERIFY ROOM IN FRMLST???????????????????
A5B9 A0 0F                  LDY     #15
A5BB B9 B6 00       LOOP95  LDA     CURFRM,Y
A5BE 91 88                  STA     (MEMA),Y
A5C0 99 C6 00               STA     PRVFRM,Y
A5C3 88                     DEY
A5C4 10 F5                  BPL     LOOP95
                        ; UPDATE COUNTERS ETC.
A5C6 EE 41 0C               INC     MOVLEN
                        ; TEST FOR OVERFLOW!!!!!!!!!!!!!!!!!!
A5C9 20 9D AE       ADDLP   JSR     WFVBFG
A5CC 20 D2 B1               JSR     CURSOR
A5CF 20 33 A3               JSR     RDKEY
A5D2 F0 03                  BEQ     EXTADD
A5D4 4C C9 A5               JMP     ADDLP
A5D7 A2 00          EXTADD  LDX     #0
A5D9 60                     RTS
```

```
                                ; CHGFRM
                                ;
A5DA AC 0C 08       CHGFRM  LDY     FRMCTR
A5DD C0 FF                  CPY     #$FF
A5DF D0 05                  BNE     *+7
A5E1 A0 00                  LDY     #0
A5E3 8C 0C 08               STY     FRMCTR
A5E6 20 49 AF               JSR     FRMACC
A5E9 A0 0F                  LDY     #15
A5EB B9 B6 00       LOOP94  LDA     CURFRM,Y
A5EE 91 B4                  STA     (PTR),Y
A5F0 88                     DEY
A5F1 10 F8                  BPL     LOOP94
A5F3 20 9D AE       CHGLP   JSR     WFVBFG
A5F6 A2 00                  LDX     #0
U   A5F8 8E 00 00           STX     STIKNO
A5FB 20 D2 B1               JSR     CURSOR
A5FE 20 33 A3               JSR     RDKEY
A601 F0 03                  BEQ     EXTCHG
A603 4C F3 A5               JMP     CHGLP
A606 A2 00          EXTCHG  LDX     #0
A608 60                     RTS
```

```
                        ; FRMSEQ
                        ;
                        ; SAVE CURFRM TO PRVFRM
A609 AE 1D 08   FRMSEQ  LDX     OLDVMO
A60C 20 C5 B2           JSR     ERSMSL
A60F A9 00              LDA     #0      RESET FRMCTR
A611 8D 0C 08   SVCRFM  STA     FRMCTR
A614 A0 0F              LDY     #15
A616 B9 B6 00   XFER11  LDA     CURFRM,Y
A619 99 C6 00           STA     PRVFRM,Y
A61C 88                 DEY
A61D 10 F7              BPL     XFER11
A61F AC 0C 08           LDY     FRMCTR
A622 20 49 AF           JSR     FRMACC
                        ; XFER FROM FRMLST TO CURFRM
A625 A0 0F              LDY     #15
A627 B1 B4      LOOP93  LDA     (PTR),Y
A629 99 B6 00           STA     CURFRM,Y
A62C 88                 DEY
A62D 10 F8              BPL     LOOP93
                        ; SET UP ??????????????????
A62F 20 77 AF   SHOWFM  JSR     BLDSCN
                        ; SET UP FRMPTR
A632 20 9D AE   SAMEFM  JSR     WFVBFG
A635 20 33 A3           JSR     RDKEY
A638 F0 26              BEQ     ENDFSQ
A63A C0 02              CPY     #2
A63C F0 03              BEQ     SUBKYS
A63E 4C 32 A6           JMP     SAMEFM  NOKEY

A641 AD 06 08   SUBKYS  LDA     SAVKEY
A644 C9 20              CMP     #'      SPACE FOR NEXT FRAME ???????????
A646 D0 10              BNE     ILSKYS
A648 AD 0C 08           LDA     FRMCTR
A64B 18                 CLC
A64C 69 01              ADC     #1
A64E CD 41 0C           CMP     MOVLEN
A651 90 02              BCC     *+4
A653 A9 00              LDA     #0
A655 4C 11 A6           JMP     SVCRFM
A658 A0 0E      ILSKYS  LDY     #14
A65A 20 CC AE           JSR     MESAGE
A65D 4C 32 A6           JMP     SAMEFM
A660 AD 15 08   ENDFSQ  LDA     HORMO
A663 AE 25 08           LDX     VPOSMO
A666 20 B6 B2           JSR     WRTMSL
A669 A2 00              LDX     #0
A66B 60                 RTS
                        ; SUBROUTINE
```

```
                                ; LOCPLR
                                ;
A66C 20 64 AF       LOCPLR  JSR     CSCNBD
A66F A2 03                  LDX     #3
A671 8E 35 08               STX     CURPLR
A674 B5 BA          NLPLR   LDA     CURFRM+PO,X
A676 C9 FF                  CMP     #$FF
A678 D0 08                  BNE     LOC1
A67A CA                     DEX
A67B 10 F7                  BPL     NLPLR
A67D A2 03                  LDX     #3
A67F 8E 35 08               STX     CURPLR
A682 4C 74 A6               JMP     NLPLR
A685 A8             LOC1    TAY
A686 20 2E AF               JSR     OBJACC
A689 AE 35 08               LDX     CURPLR
A68C A0 00                  LDY     #0
A68E B1 B4                  LDA     (PTR),Y
A690 85 8C                  STA     MSP
A692 C8                     INY
A693 B1 B4                  LDA     (PTR),Y
A695 85 8D                  STA     MSP+1
A697 A0 03                  LDY     #DATLEN
A699 B1 8C                  LDA     (MSP),Y
A69B 9D 29 08               STA     HEIGHT,X
A69E 9D 2D 08               STA     OLDHGT,X
A6A1 B5 BE                  LDA     CURFRM+XO,X
A6A3 8D 15 08               STA     HORMO
A6A6 B5 C2                  LDA     CURFRM+YO,X
A6A8 8D 25 08               STA     VPOSMO
A6AB 9D 19 08               STA     OLDVPO,X
A6AE 20 9D AE       LOCLP   JSR     WFVBFG
A6B1 20 33 A3               JSR     RDKEY
A6B4 F0 31                  BEQ     LEXIT2  VALID EXIT
A6B6 C0 02                  CPY     #2
A6B8 F0 03                  BEQ     SUBKYL
A6BA 4C F7 A6               JMP     MOVEIT
A6BD AD 06 08       SUBKYL  LDA     SAVKEY
A6C0 C9 20                  CMP     #'      SPACE KEY?????????
A6C2 D0 1B                  BNE     ILSKYL
A6C4 AE 35 08               LDX     CURPLR
A6C7 BD 11 08               LDA     HORPO,X
A6CA 95 BE                  STA     CURFRM+XO,X
A6CC BD 21 08               LDA     VPOSPO,X
A6CF 95 C2                  STA     CURFRM+YO,X
A6D1 CA                     DEX
A6D2 CE 35 08               DEC     CURPLR
A6D5 10 9D                  BPL     NLPLR
A6D7 A2 03                  LDX     #3
A6D9 8E 35 08               STX     CURPLR
A6DC 4C 74 A6               JMP     NLPLR
A6DF A0 0E          ILSKYL  LDY     #14
A6E1 20 CC AE               JSR     MESAGE
A6E4 4C F7 A6               JMP     MOVEIT
A6E7 AE 35 08       LEXIT2  LDX     CURPLR
A6EA BD 11 08               LDA     HORPO,X
A6ED 95 BE                  STA     CURFRM+XO,X
A6EF BD 21 08               LDA     VPOSPO,X
A6F2 95 C2                  STA     CURFRM+YO,X
A6F4 A2 00                  LDX     #0
```

```
A6F6 60                 RTS
                ; MOVE CURSOR
A6F7 20 D2 B1   MOVEIT  JSR     CURSOR
                ; UPDATE PLAYER POSITION WITH CURSOR POSITION
A6FA AE 35 08           LDX     CURPLR
A6FD AD 15 08           LDA     HORMO
A700 9D 11 08           STA     HORPO,X
A703 18                 CLC
A704 69 30              ADC     #48
A706 9D 00 D0           STA     HPOSPO,X
A709 AD 25 08           LDA     VPOSMO
A70C 9D 21 08           STA     VPOSPO,X
                ; SET UP SOURCE POINTER
A70F AD 35 08           LDA     CURPLR
A712 0A                 ASL     A
A713 A8                 TAY
A714 B9 EC AC           LDA     PLYRTB,Y
A717 8D 68 08           STA     TMP1
                ;
                ;
A71A B9 ED AC           LDA     PLYRTB+1,Y
A71D 8D 69 08           STA     TMP2
A720 85 8D              STA     MSP+1
A722 AE 35 08           LDX     CURPLR
A725 AD 68 08           LDA     TMP1
A728 18                 CLC
A729 7D 19 08           ADC     OLDVPO,X
A72C 85 8C              STA     MSP
A72E 90 02              BCC     *+4
A730 E6 8D              INC     MSP+1
A732 AD 69 08           LDA     TMP2
A735 85 8F              STA     MDP+1
A737 AD 68 08           LDA     TMP1
A73A 18                 CLC
A73B 7D 21 08           ADC     VPOSPO,X
A73E 85 8E              STA     MDP
A740 90 02              BCC     *+4
A742 E6 8F              INC     MDP+1
A744 BD 29 08   LOC222  LDA     HEIGHT,X
A747 85 90      LOC333  STA     MBC
A749 A9 00              LDA     #0
A74B 85 91              STA     MBC+1
A74D BD 21 08           LDA     VPOSPO,X
A750 DD 19 08           CMP     OLDVPO,X
A753 90 24              BCC     MOVEUP
                ; MOVE PLAYER DOWN
A755 20 84 B4           JSR     MOVDA
A758 AE 35 08           LDX     CURPLR
A75B A5 8C              LDA     MSP
A75D 85 8E              STA     MDP
A75F A5 8D              LDA     MSP+1
A761 85 8F              STA     MDP+1
A763 BD 21 08           LDA     VPOSPO,X
A766 38                 SEC
A767 FD 19 08           SBC     OLDVPO,X
A76A 85 90              STA     MBC
A76C A9 00              LDA     #0
A76E 85 91              STA     MBC+1
A770 8D 67 08           STA     FILVAL
A773 20 7B AD           JSR     FIL
```

```
A776 4C 9E A7                JMP     LOC123
                        ; MOVE PLAYER UP
A779 20 5F B4       MOVEUP   JSR     MOVIA
A77C AE 35 08                LDX     CURPLR
A77F A5 8E                   LDA     MDP
A781 18                      CLC
A782 7D 29 08                ADC     HEIGHT,X
A785 85 8E                   STA     MDP
A787 90 02                   BCC     *+4
A789 E6 8F                   INC     MDP+1
A78B BD 19 08                LDA     OLDVPO,X
A78E 38                      SEC
A78F FD 21 08                SBC     VPOSPO,X
A792 85 90                   STA     MBC
A794 A9 00                   LDA     #0
A796 85 91                   STA     MBC+1
A798 8D 67 08                STA     FILVAL
A79B 20 7B AD                JSR     FIL
                        ; UPDATE OLD VPOS AND HEIGHT
A79E BD 21 08       LOC123   LDA     VPOSPO,X
A7A1 9D 19 08                STA     OLDVPO,X
A7A4 BD 29 08                LDA     HEIGHT,X
A7A7 9D 2D 08                STA     OLDHGT,X
A7AA 4C AE A6                JMP     LOCLP
```

```
                        ; RUN
                        ;
A7AD AE 1D 08   RUN     LDX     OLDVMO
A7B0 20 C5 B2           JSR     ERSMSL
A7B3 A9 00              LDA     #0
A7B5 8D 0C 08   RUNLP   STA     FRMCTR
A7B8 A8                 TAY
A7B9 20 49 AF           JSR     FRMACC
A7BC A0 0F              LDY     #15
A7BE B1 B4      LOOP92  LDA     (PTR),Y
A7C0 99 B6 00           STA     CURFRM,Y
A7C3 88                 DEY
A7C4 10 F8              BPL     LOOP92
A7C6 A0 04              LDY     #4      MULTI FRAME DELAY
A7C8 20 9D AE   RUNSNC  JSR     WFVBFG
A7CB 88                 DEY
A7CC 10 FA              BPL     RUNSNC
A7CE AD FC 02           LDA     CH
A7D1 C9 FF              CMP     #$FF
A7D3 F0 13              BEQ     BUILD
A7D5 20 71 B0           JSR     GETKEE
A7D8 C9 9B              CMP     #$9B
A7DA D0 0C              BNE     BUILD
A7DC AD 15 08           LDA     HORMO
A7DF AE 25 08           LDX     VPOSMO
A7E2 20 B6 B2           JSR     WRTMSL
A7E5 A2 00              LDX     #0
A7E7 60                 RTS
A7E8 20 77 AF   BUILD   JSR     BLDSCN
A7EB AD 0C 08           LDA     FRMCTR
A7EE 18                 CLC
A7EF 69 01              ADC     #1
A7F1 CD 41 0C           CMP     MOVLEN
A7F4 90 BF              BCC     RUNLP
A7F6 A0 7F              LDY     #$7F    DELAY
A7F8 20 9D AE   RLP1    JSR     WFVBFG
A7FB 88                 DEY
A7FC 10 FA              BPL     RLP1
A7FE A9 00              LDA     #0
A800 4C B5 A7           JMP     RUNLP
```

```
                        ; MODPLR
                        ;
A803 A2 03      MODPLR  LDX     #3
A805 8E 35 08           STX     CURPLR
A808 A9 01              LDA     #1
A80A 8D 07 08           STA     CLSLCT
A80D 20 64 AF   MSTPBD  JSR     CSCNBD
A810 A9 00              LDA     #0
A812 8D 04 08           STA     CHGFLG
A815 B5 BA              LDA     CURFRM+PO,X
A817 C9 FF              CMP     #$FF
A819 D0 03              BNE     *+5
A81B 4C 74 A8           JMP     TRYNXT
                ; IF PLAYER AVAIL, SET CURSOR.  IF NOT, GO TO NEXT PLAYER.
A81E B5 BE              LDA     CURFRM+XO,X
A820 8D 15 08           STA     HORMO
A823 9D 11 08           STA     HORPO,X
A826 18                 CLC
A827 69 30              ADC     #48
A829 9D 00 D0           STA     HPOSPO,X
A82C B5 C2              LDA     CURFRM+YO,X
A82E 8D 25 08           STA     VPOSMO
A831 9D 21 08           STA     VPOSPO,X
A834 20 A8 AE           JSR     CPLRBD
                ; CHECK COLOR SELECT, CONT., AND RETURN KEYS
A837 20 9D AE   MODLP   JSR     WFVBFQ
A83A AD FC 02           LDA     CH
A83D C9 FF              CMP     #$FF
A83F D0 03              BNE     *+5
A841 4C 95 A8           JMP     NOKEY2
A844 20 71 B0           JSR     GETKEE
A847 C9 30              CMP     #'0
A849 D0 08              BNE     MCHK1
A84B A9 00              LDA     #0
A84D 8D 07 08           STA     CLSLCT
A850 4C 95 A8           JMP     NOKEY2
A853 C9 31      MCHK1   CMP     #'1
A855 D0 08              BNE     MCHKM
A857 A9 01              LDA     #1
A859 8D 07 08           STA     CLSLCT
A85C 4C 95 A8           JMP     NOKEY2
A85F C9 4D      MCHKM   CMP     #'M
A861 F0 03              BEQ     *+5
A863 4C 7F A8           JMP     MCHKRT
                ; IF ANY CHANGE, SAVE TO NEW PDATA AND GOTO NEXT PLAYER
A866 AD 04 08           LDA     CHGFLG
A869 F0 09              BEQ     TRYNXT
A86B AD 40 0C           LDA     OLSTLN
A86E 8D 0D 08           STA     OBJCTR
A871 20 01 AD           JSR     SAVALL
A874 CE 35 08   TRYNXT  DEC     CURPLR
A877 30 19              BMI     ENDMOD
A879 AE 35 08           LDX     CURPLR
A87C 4C 0D A8           JMP     MSTPBD
A87F C9 9B      MCHKRT  CMP     #$9B
A881 D0 12              BNE     NOKEY2
                ; RETURN
A883 AD 04 08           LDA     CHGFLG
A886 F0 0A              BEQ     ENDMOD  NO CHANGES, EXIT.
                ; CHANGE MADEE, SAVE TO NEW PDATA
```

```
A88B AD 40 0C            LDA    OLSTLN
A88B 8D 0D 08            STA    OBJCTR
A88E 20 01 AD            JSR    SAVALL
A891 60                  RTS
A892 A2 00       ENDMOD  LDX    #0
A894 60                  RTS
A895 20 D2 B1    NOKEY2  JSR    CURSOR
A898 A2 00               LDX    #0
A89A BD 84 02            LDA    STRIGO,X
A89D D0 0B               BNE    KEEPON
A89F A9 01               LDA    #1
A8A1 8D 04 08            STA    CHGFLG
A8A4 AE 35 08            LDX    CURPLR
A8A7 20 76 AE            JSR    INKPOS
A8AA 4C 37 A8    KEEPON  JMP    MODLP
```

```
                        ; CASSAV

                        ; OPEN CASSETTE FOR WRITE
A8AD A2 30              CASSAV  LDX     #$30    IOCB3
A8AF A9 03                      LDA     #OPEN
A8B1 9D 42 03                   STA     ICCOM,X
A8B4 A9 08                      LDA     #OPNOT  OUTPUT
A8B6 9D 4A 03                   STA     ICAX1,X
A8B9 A9 00                      LDA     #0      NORMAL IRG
A8BB 9D 4B 03                   STA     ICAX2,X
A8BE A9 B4                      LDA     #CASS
A8C0 9D 44 03                   STA     ICBAL,X
A8C3 A9 A2                      LDA     #CASS/256
A8C5 9D 45 03                   STA     ICBAH,X
A8C8 20 56 E4                   JSR     CIOV
                        ; STATUS?????????
                        ; SAVE MEM MGMNT PTRS
A8CB A9 10                      LDA     #16
A8CD 85 90                      STA     MBC
A8CF A9 00                      LDA     #0
A8D1 85 91                      STA     MBC+1
A8D3 A2 30                      LDX     #$30
A8D5 A9 0B                      LDA     #PUTCHR
A8D7 9D 42 03                   STA     ICCOM,X
A8DA A9 90                      LDA     #MBC
A8DC 9D 44 03                   STA     ICBAL,X
A8DF A9 00                      LDA     #MBC/256
A8E1 9D 45 03                   STA     ICBAH,X
A8E4 A9 02                      LDA     #2
A8E6 9D 48 03                   STA     ICBLL,X
A8E9 A9 00                      LDA     #0
A8EB 9D 49 03                   STA     ICBLH,X
A8EE 20 56 E4                   JSR     CIOV
                        ;
A8F1 A2 30                      LDX     #$30
A8F3 A9 0B                      LDA     #PUTCHR
A8F5 9D 42 03                   STA     ICCOM,X
A8F8 A9 94                      LDA     #LS1L
A8FA 9D 44 03                   STA     ICBAL,X
A8FD A9 00                      LDA     #LS1L/256
A8FF 9D 45 03                   STA     ICBAH,X
A902 A5 90                      LDA     MBC
A904 9D 48 03                   STA     ICBLL,X
A907 A5 91                      LDA     MBC+1
A909 9D 49 03                   STA     ICBLH,X
A90C 20 56 E4                   JSR     CIOV
                        ; WRITE BYTE COUNT (2 BYTE VARIABLE)
                        ; CALC BYTE COUNT FROM LS1L TO DS2H
A90F A5 9E                      LDA     DS1H
A911 85 92                      STA     ACC
A913 A5 9F                      LDA     DS1H+1
A915 85 93                      STA     ACC+1
A917 A0 14                      LDY     #LS1L-DTAB
                        ; ACC = ACC - LS1L
A919 20 25 B3                   JSR     DSUBA
                        ; STORE BYTE COUNT
A91C A2 30                      LDX     #$30
A91E A9 0B                      LDA     #PUTCHR
A920 9D 42 03                   STA     ICCOM,X
A923 A9 92                      LDA     #ACC
A925 9D 44 03                   STA     ICBAL,X
```

```
A928 A9 00          LDA     #ACC/256
A92A 9D 45 03       STA     ICBAH,X
A92D A9 02          LDA     #2
A92F 9D 48 03       STA     ICBLL,X
A932 A9 00          LDA     #0
A934 9D 49 03       STA     ICBLH,X
A937 20 56 E4       JSR     CIOV
                    ; STATUS???????
                    ; WRITE DATA
A93A A2 30          LDX     #$30
A93C A9 08          LDA     #PUTCHR
A93E 9D 42 03       STA     ICCOM,X
A941 A9 6F          LDA     #FRMLST
A943 9D 44 03       STA     ICBAL,X
A946 A9 08          LDA     #FRMLST/256
A948 9D 45 03       STA     ICBAH,X
A94B A5 92          LDA     ACC
A94D 9D 48 03       STA     ICBLL,X
A950 A5 93          LDA     ACC+1
A952 9D 49 03       STA     ICBLH,X
A955 20 56 E4       JSR     CIOV
                    ; STATUS?????????
                    ; STORE BYTE COUNT
A958 A5 A2          LDA     DS2H
A95A 85 92          STA     ACC
A95C A5 A3          LDA     DS2H+1
A95E 85 93          STA     ACC+1
A960 A0 20          LDY     #DS2L-DTAB
A962 20 25 B3       JSR     DSUBA
                    ;
                    ; STORE BYTE COUNT
A965 A2 30          LDX     #$30
A967 A9 0B          LDA     #PUTCHR
A969 9D 42 03       STA     ICCOM,X
A96C A9 92          LDA     #ACC
A96E 9D 44 03       STA     ICBAL,X
A971 A9 00          LDA     #ACC/256
A973 9D 45 03       STA     ICBAH,X
A976 A9 02          LDA     #2
A978 9D 48 03       STA     ICBLL,X
A97B A9 00          LDA     #0
A97D 9D 49 03       STA     ICBLH,X
A980 20 56 E4       JSR     CIOV
                    ; WRITE DATA
A983 A2 30          LDX     #$30
A985 A9 0B          LDA     #PUTCHR
A987 9D 42 03       STA     ICCOM,X
A98A A5 A0          LDA     DS2L
A98C 9D 44 03       STA     ICBAL,X
A98F A5 A1          LDA     DS2L+1
A991 9D 45 03       STA     ICBAH,X
A994 A5 92          LDA     ACC
A996 9D 48 03       STA     ICBLL,X
A999 A5 93          LDA     ACC+1
A99B 9D 49 03       STA     ICBLH,X
A99E 20 56 E4       JSR     CIOV
                    ; STATUS???????????
                    ; CLOSE CASSETTE
A9A1 A2 30          LDX     #$30
A9A3 A9 0C          LDA     #CLOSE
```

```
A9A5 9D 42 03           STA     ICCOM,X
A9A8 20 56 E4           JSR     CIOV
                ; STATUS???????
A9AB A2 00              LDX     #0
A9AD 60                 RTS
```

```
                        ; CASSRD

                        ; OPEN CASSETTE FOR READ
A9AE A2 30      CASSRD  LDX     #$30    IOCB3
A9B0 A9 03              LDA     #OPEN
A9B2 9D 42 03           STA     ICCOM,X
A9B5 A9 04              LDA     #OPNIN  INPUT
A9B7 9D 4A 03           STA     ICAX1,X
A9BA A9 00              LDA     #0      NORMAL IRG
A9BC 9D 4B 03           STA     ICAX2,X
A9BF A9 B4              LDA     #CASS
A9C1 9D 44 03           STA     ICBAL,X
A9C4 A9 A2              LDA     #CASS/256
A9C6 9D 45 03           STA     ICBAH,X
A9C9 20 56 E4           JSR     CIOV
                        ; STATUS????????
                        ; READ BYTE COUNT FROM TAPE
A9CC A2 30              LDX     #$30
A9CE A9 07              LDA     #GETCHR
A9D0 9D 42 03           STA     ICCOM,X
A9D3 A9 90              LDA     #MBC
A9D5 9D 44 03           STA     ICBAL,X
A9D8 A9 00              LDA     #MBC/256
A9DA 9D 45 03           STA     ICBAH,X
A9DD A9 02              LDA     #2
A9DF 9D 48 03           STA     ICBLL,X
A9E2 A9 00              LDA     #0
A9E4 9D 49 03           STA     ICBLH,X
A9E7 20 56 E4           JSR     CIOV
                        ;
A9EA A2 30              LDX     #$30
A9EC A9 07              LDA     #GETCHR
A9EE 9D 42 03           STA     ICCOM,X
A9F1 A9 94              LDA     #LS1L
A9F3 9D 44 03           STA     ICBAL,X
A9F6 A9 00              LDA     #LS1L/256
A9F8 9D 45 03           STA     ICBAH,X
A9FB A5 90              LDA     MBC
A9FD 9D 48 03           STA     ICBLL,X
AA00 A5 91              LDA     MBC+1
AA02 9D 49 03           STA     ICBLH,X
AA05 20 56 E4           JSR     CIOV
                        ; READ BYTE COUNT FROM TAPE
AA08 A2 30              LDX     #$30
AA0A A9 07              LDA     #GETCHR
AA0C 9D 42 03           STA     ICCOM,X
AA0F A9 90              LDA     #MBC
AA11 9D 44 03           STA     ICBAL,X
AA14 A9 00              LDA     #MBC/256
AA16 9D 45 03           STA     ICBAH,X
AA19 A9 02              LDA     #2
AA1B 9D 48 03           STA     ICBLL,X
AA1E A9 00              LDA     #0
AA20 9D 49 03           STA     ICBLH,X
AA23 20 56 E4           JSR     CIOV
                        ; STATUS????????
                        ; READ DATA
AA26 A2 30              LDX     #$30
AA28 A9 07              LDA     #GETCHR
AA2A 9D 42 03           STA     ICCOM,X
AA2D A9 6F              LDA     #FRMLST
```

```
AA2F 9D 44 03            STA     ICBAL,X
AA32 A9 08               LDA     #FRMLST/256
AA34 9D 45 03            STA     ICBAH,X
AA37 A5 90               LDA     MBC
AA39 9D 48 03            STA     ICBLL,X
AA3C A5 91               LDA     MBC+1
AA3E 9D 49 03            STA     ICBLH,X
AA41 20 56 E4            JSR     CIOV
                    ; STATUS?????????
                    ; READ BYTE COUNT FROM TAPE
AA44 A2 30               LDX     #$30
AA46 A9 07               LDA     #GETCHR
AA48 9D 42 03            STA     ICCOM,X
AA4B A9 90               LDA     #MBC
AA4D 9D 44 03            STA     ICBAL,X
AA50 A9 00               LDA     #MBC/256
AA52 9D 45 03            STA     ICBAH,X
AA55 A9 02               LDA     #2
AA57 9D 48 03            STA     ICBLL,X
AA5A A9 00               LDA     #0
AA5C 9D 49 03            STA     ICBLH,X
AA5F 20 56 E4            JSR     CIOV
AA62 A5 A2               LDA     DS2H
AA64 85 92               STA     ACC
AA66 A5 A3               LDA     DS2H+1
AA68 85 93               STA     ACC+1
AA6A A0 10               LDY     #MBC-DTAB
AA6C 20 25 B3            JSR     DSUBA
AA6F A2 30               LDX     #$30
AA71 A9 07               LDA     #GETCHR
AA73 A5 92               LDA     ACC
AA75 9D 44 03            STA     ICBAL,X
AA78 A5 93               LDA     ACC+1
AA7A 9D 45 03            STA     ICBAH,X
AA7D A5 90               LDA     MBC
AA7F 9D 48 03            STA     ICBLL,X
AA82 A5 91               LDA     MBC+1
AA84 9D 49 03            STA     ICBLH,X
AA87 20 56 E4            JSR     CIOV
                    ; STATUS???????????
                    ; CLOSE CASSETTE
AA8A A2 30               LDX     #$30
AA8C A9 0C               LDA     #CLOSE
AA8E 9D 42 03            STA     ICCOM,X
AA91 20 56 E4            JSR     CIOV
                    ; STATUS????????
AA94 A2 00               LDX     #0
AA96 60                  RTS
```

```
                        ; ZERMEM
                        ; ZERO PLAYERS
                        ;
AA97 A9 00      ZERMEM  LDA     #0
AA99 8D 67 08           STA     FILVAL
AA9C A9 80              LDA     #$80
AA9E 85 90              STA     MBC
AAA0 A9 02              LDA     #2
AAA2 85 91              STA     MBC+1
AAA4 A9 80              LDA     #GXMISL
AAA6 85 8E              STA     MDP
AAA8 A9 05              LDA     #GXMISL/256
AAAA 85 8F              STA     MDP+1
AAAC 20 7B AD           JSR     FIL
                        ; SET UP S POINTERS --- REINIT
                        ;
AAAF A9 2F              LDA     #BACLST
AAB1 85 86              STA     S2H
AAB3 A9 0C              LDA     #BACLST/256
AAB5 85 87              STA     S2H+1
AAB7 A9 6F              LDA     #FRMLST
AAB9 8D E7 02           STA     MEMLO
AABC 85 80              STA     S1L
AABE A9 08              LDA     #FRMLST/256
AAC0 8D E8 02           STA     MEMLO+1
AAC3 85 81              STA     S1L+1
AAC5 20 2F B3           JSR     MINIT
AAC8 20 0A AF           JSR     S1TLS1
                        ;
AACB A9 42              LDA     #BACLST+19
AACD 85 80              STA     S1L
AACF A9 0C              LDA     #BACLST+19/256
AAD1 85 81              STA     S1L+1
AAD3 AD E5 02           LDA     MEMTOP
AAD6 85 86              STA     S2H
AAD8 AD E6 02           LDA     MEMTOP+1
AADB 85 87              STA     S2H+1
AADD 20 2F B3           JSR     MINIT   INITS S1H AND S2L - USE REGION 2
AAE0 20 22 AF           JSR     S1TDS1
                        ; ZERO FROM FRMLST TO DS2H
AAE3 A9 6F              LDA     #FRMLST
AAE5 85 8E              STA     MDP
AAE7 A9 08              LDA     #FRMLST/256
AAE9 85 8F              STA     MDP+1
                        ; CALC BYTE COUNT FROM LS1L TO DS2H
AAEB A5 A2              LDA     DS2H
AAED 85 92              STA     ACC
AAEF A5 A3              LDA     DS2H+1
AAF1 85 93              STA     ACC+1
AAF3 A0 14              LDY     #LS1L-DTAB
                        ; ACC = ACC - LS1L
AAF5 20 25 B3           JSR     DSUBA
                        ; STORE BYTE COUNT
AAF8 A5 92              LDA     ACC
AAFA 85 90              STA     MBC
AAFC A5 93              LDA     ACC+1
AAFE 85 91              STA     MBC+1
AB00 A9 00              LDA     #0
AB02 8D 67 08           STA     FILVAL
AB05 20 7B AD           JSR     FIL
```

```
                        ; SET POINTER TO BLANK BACKGROUND INTO BACLST
AB08 AC 3F 0C                LDY    BLSTLN
AB0B 20 A2 AD                JSR    BACACC
AB0E EE 3F 0C                INC    BLSTLN
AB11 A0 01                   LDY    #1
AB13 B9 A8 00      LOOP90    LDA    BLNKPT,Y
AB16 91 B4                   STA    (PTR),Y
AB18 88                      DEY
AB19 10 FB                   BPL    LOOP90
AB1B A0 0F                   LDY    #FELELN-1
AB1D A9 FF                   LDA    #$FF
AB1F 99 B6 00      LOOP89    STA    CURFRM,Y
AB22 99 C6 00                STA    PRVFRM,Y
AB25 88                      DEY
AB26 10 F7                   BPL    LOOP89
AB28 A2 00                   LDX    #0
AB2A 60                      RTS
```

```
                        ; DELBAK
                        ; DEALLOCATE BDATA
AB2B A0 03              DELBAK  LDY     #BAKGND
AB2D B9 B6 00                   LDA     CURFRM,Y
AB30 8D 0E 08                   STA     BACCTR
AB33 D0 03                      BNE     *+5
AB35 A2 06                      LDX     #6      CANT DELETE BLANK BACK
AB37 60                         RTS
AB38 A8                         TAY
AB39 20 A2 AD                   JSR     BACACC
AB3C A0 00                      LDY     #0
AB3E B1 B4                      LDA     (PTR),Y
AB40 85 8E                      STA     MDP
AB42 C8                         INY
AB43 B1 B4                      LDA     (PTR),Y
AB45 85 8F                      STA     MDP+1
                        ;
AB47 A0 00                      LDY     #0
AB49 B1 8E                      LDA     (MDP),Y
AB4B 85 90                      STA     MBC
AB4D 85 A4                      STA     PUPDAT
AB4F C8                         INY
AB50 B1 8E                      LDA     (MDP),Y
AB52 85 91                      STA     MBC+1
AB54 85 A5                      STA     PUPDAT+1
                        ;
AB56 A5 8E                      LDA     MDP
AB58 85 88                      STA     MEMA
AB5A A5 8F                      LDA     MDP+1
AB5C 85 89                      STA     MEMA+1
AB5E 20 16 AF                   JSR     DS1TS1
AB61 20 D9 B3                   JSR     MDEALL
AB64 20 22 AF                   JSR     S1TDS1
                        ; DO MEM AVAIL MESSAGE HERE
                        ; DEALLOCATE BACLST PTR
AB67 AD 3F 0C                   LDA     BLSTLN
AB6A 38                         SEC
AB6B ED 0E 08                   SBC     BACCTR
AB6E 38                         SEC
AB6F E9 01                      SBC     #1
AB71 8D 68 08                   STA     TMP1
AB74 0A                         ASL     A
AB75 85 90                      STA     MBC
AB77 A9 00                      LDA     #0
AB79 85 91                      STA     MBC+1
AB7B A5 B5                      LDA     PTR+1
AB7D 85 8F                      STA     MDP+1
AB7F 85 8D                      STA     MSP+1
AB81 A5 B4                      LDA     PTR
AB83 85 8E                      STA     MDP
AB85 18                         CLC
AB86 69 02                      ADC     #2
AB88 85 8C                      STA     MSP
AB8A 90 02                      BCC     *+4
AB8C E6 8D                      INC     MSP+1
AB8E 20 5F B4                   JSR     MOVIA
AB91 CE 3F 0C                   DEC     BLSTLN
                        ; UPDATE BACLST POINTERS
AB94 A0 00              UPDTBP  LDY     #0
AB96 B1 B4                      LDA     (PTR),Y
```

```
AB98 85 92                  STA     ACC
AB9A C8                     INY
AB9B B1 B4                  LDA     (PTR),Y
AB9D 85 93                  STA     ACC+1
AB9F A0 24                  LDY     #PUPDAT-DTAB
ABA1 20 20 B3               JSR     DADDA
                    ;
ABA4 A0 00                  LDY     #0
ABA6 A5 92                  LDA     ACC
ABA8 91 B4                  STA     (PTR),Y
ABAA C8                     INY
ABAB A5 93                  LDA     ACC+1
ABAD 91 B4                  STA     (PTR),Y
ABAF CE 68 08               DEC     TMP1
ABB2 F0 0E                  BEQ     UPDTRF
ABB4 A5 B4                  LDA     PTR
ABB6 18                     CLC
ABB7 69 02                  ADC     #2
ABB9 85 B4                  STA     PTR
ABBB 90 02                  BCC     *+4
ABBD E6 B5                  INC     PTR+1
ABBF 4C 94 AB               JMP     UPDTBP
                    ; UPDATE ALL LIST REFS
ABC2 A0 40          UPDTRF  LDY     #MOVLEN-1
ABC4 8C 0C 08               STY     FRMCTR
ABC7 20 49 AF       XXXX    JSR     FRMACC
ABCA A0 03                  LDY     #BAKGND
ABCC B1 B4                  LDA     (PTR),Y
ABCE 30 13                  BMI     YYYY
ABD0 CD 0E 08               CMP     BACCTR
ABD3 F0 0A                  BEQ     SETFF
ABD5 90 0C                  BCC     YYYY
ABD7 38                     SEC
ABD8 E9 01                  SBC     #1
ABDA 91 B4                  STA     (PTR),Y
ABDC 4C E3 AB               JMP     YYYY
ABDF A9 FF          SETFF   LDA     #$FF
ABE1 91 B4                  STA     (PTR),Y
ABE3 CE 0C 08       YYYY    DEC     FRMCTR
ABE6 AC 0C 08               LDY     FRMCTR
ABE9 10 DC                  BPL     XXXX
                    ; AND DO FOR CURFRM,PRVFRM ALSO!!!!!!!!!!
ABEB A0 03                  LDY     #BAKGND
ABED A9 FF                  LDA     #$FF
ABEF 99 B6 00               STA     CURFRM,Y
ABF2 A2 00                  LDX     #0
ABF4 60                     RTS
```

```
                                ; BACSEQ
                                ; MOVE PLAYERS OFF
ABF5 A2 03          BACSEQ  LDX     #3
ABF7 A9 A8          POFFLP  LDA     #PRTEND+8
ABF9 9D 11 08               STA     HORPO,X
ABFC 18                     CLC
ABFD 69 30                  ADC     #48
ABFF 9D 00 D0               STA     HPOSPO,X
AC02 CA                     DEX
AC03 10 F2                  BPL     POFFLP
                                ; DO BACK SEQUENCING
AC05 A9 01                  LDA     #1
AC07 8D 0E 08               STA     BACCTR
AC0A 4C 32 AC               JMP     SHOWB
AC0D 20 9D AE       BSLP    JSR     WFVBFG
AC10 AD FC 02               LDA     CH
AC13 C9 FF                  CMP     #$FF
AC15 F0 1B                  BEQ     SHOWB
AC17 20 71 B0               JSR     GETKEE
AC1A C9 9B                  CMP     #$9B
AC1C F0 43                  BEQ     ENDBSQ
AC1E C9 56                  CMP     #'V
AC20 D0 10                  BNE     SHOWB
AC22 EE 0E 08               INC     BACCTR
AC25 AD 0E 08               LDA     BACCTR
AC28 CD 3F 0C               CMP     BLSTLN
AC2B D0 05                  BNE     SHOWB
AC2D A9 01                  LDA     #1
AC2F 8D 0E 08               STA     BACCTR
AC32 AC 0E 08       SHOWB   LDY     BACCTR
AC35 20 A2 AD               JSR     BACACC
AC38 A0 00                  LDY     #0
AC3A B1 B4                  LDA     (PTR),Y
AC3C 8D 68 08               STA     TMP1
AC3F C8                     INY
AC40 B1 B4                  LDA     (PTR),Y
AC42 8D 69 08               STA     TMP2
AC45 A0 04                  LDY     #4
AC47 AD 68 08               LDA     TMP1
AC4A 18                     CLC
AC4B 69 02                  ADC     #2
AC4D 91 AE                  STA     (DLPTR),Y
AC4F 85 AC                  STA     RAMPTR
AC51 90 03                  BCC     *+5
AC53 EE 69 08               INC     TMP2
AC56 C8                     INY
AC57 AD 69 08               LDA     TMP2
AC5A 91 AE                  STA     (DLPTR),Y
AC5C 85 AD                  STA     RAMPTR+1
AC5E 4C 0D AC               JMP     BSLP
                    ;
AC61 A0 0F          ENDBSQ  LDY     #15
AC63 A9 FF                  LDA     #$FF
AC65 99 B6 00       LOOP88  STA     CURFRM,Y
AC68 88                     DEY
AC69 10 FA                  BPL     LOOP88
AC6B A0 03                  LDY     #BAKGND
AC6D AD 0E 08               LDA     BACCTR
AC70 99 B6 00               STA     CURFRM,Y
AC73 A2 00                  LDX     #0
```

```
AC75 60                    RTS
```

```
                                ; ERSCRN
                                ; SAVE CURFRM
AC76 A0 0F          ERSCRN  LDY     #15
AC78 B9 B6 00       LOOP80  LDA     CURFRM,Y
AC7B 99 C6 00               STA     PRVFRM,Y
AC7E 88                     DEY
AC7F 10 F7                  BPL     LOOP80
                                ; CLEAR OUT CURFRM
AC81 A0 0F                  LDY     #15
AC83 A9 FF                  LDA     #$FF
AC85 99 B6 00       LOOP79  STA     CURFRM,Y
AC88 88                     DEY
AC89 10 FA                  BPL     LOOP79
AC8B A0 03                  LDY     #BAKGND
AC8D A9 00                  LDA     #0
AC8F 99 B6 00               STA     CURFRM,Y
AC92 20 9D AE       ELOOP   JSR     WFVBFG
AC95 20 33 A3               JSR     RDKEY
AC98 F0 0F                  BEQ     EXTERS
AC9A C0 02                  CPY     #2
AC9C B0 03                  BCS     NOALWE
AC9E 4C 92 AC               JMP     ELOOP
ACA1 A0 0C          NOALWE  LDY     #12     2 X 6
ACA3 20 CC AE               JSR     MESAGE
ACA6 4C 92 AC               JMP     ELOOP
ACA9 A2 00          EXTERS  LDX     #0
ACAB 60                     RTS
```

```
                        ; SETSPD
                        ;
ACAC 20 9D AE   SETSPD  JSR     WFVBFG
ACAF 20 33 A3           JSR     RDKEY
ACB2 F0 25              BEQ     ENDSPD
ACB4 C0 02              CPY     #2
ACB6 F0 05              BEQ     SUBKYW
ACB8 B0 17              BCS     NOALWW
ACBA 4C AC AC           JMP     SETSPD
ACBD AD 06 08   SUBKYW  LDA     SAVKEY
ACC0 C9 30              CMP     #'0
ACC2 90 E8              BCC     SETSPD
ACC4 C9 40              CMP     #$40
ACC6 B0 E4              BCS     SETSPD
ACC8 38                 SEC
ACC9 E9 30              SBC     #$30
ACCB 8D 05 08           STA     MSPEED
ACCE 4C AC AC           JMP     SETSPD
ACD1 A0 0C      NOALWW  LDY     #12     2 X 6
ACD3 20 CC AE           JSR     MESAGE
ACD6 4C AC AC           JMP     SETSPD
ACD9 A2 00      ENDSPD  LDX     #0
ACDB 60                 RTS
```

```
                        ;
ACDC FE FD FB       ZERTBL  .BYTE   $FE,$FD,$FB,$F7,$EF,$DF,$BF,$7F
ACDF F7 EF DF
ACE2 BF 7F
ACE4 01 02 04       ONETBL  .BYTE   1,2,4,8,16,32,64,128
ACE7 08 10 20
ACEA 40 80
ACEC 10 06 90       PLYRTB  .WORD   GXP0+16,GXP1+16,GXP2+16,GXP3+16
ACEF 06 10 07
ACF2 90 07
                        ;               P0  P1  P2  P3  PF0 PF1 PF2 PF3 BK
ACF4 18 88 B8       COLRTB  .BYTE   $18,$88,$B8,$58
ACF7 58
                        ;
                        ;
```

```
                        ;
                        ; VBLANK INTERRUPT HANDLER
                        ;
ACF8 EA          VBHND  NOP
ACF9 A9 01             LDA    #1
ACFB 8D 4B 08          STA    VBFLAG
ACFE 4C 62 E4    EVBHND JMP    XITVBV
                        ;
                        ;
                        ;
                        ;
```

```
                        ;
                        ; SUBROUTINES
                        ;
                        ; SUBROUTINE SAVALL
AD01 20 14 AE     SAVALL  JSR     VRTSIZ
AD04 AE 35 08             LDX     CURPLR
AD07 BD 29 08             LDA     HEIGHT,X
AD0A F0 61                BEQ     ZERPLR
AD0C 18                   CLC
AD0D 69 04                ADC     #4
AD0F 85 8A                STA     MEMB
AD11 A9 00                LDA     #0
AD13 85 8B                STA     MEMB+1
AD15 20 16 AF             JSR     DS1TS1
AD18 A5 82                LDA     S1H
AD1A 85 88                STA     MEMA
AD1C A5 83                LDA     S1H+1
AD1E 85 89                STA     MEMA+1
AD20 20 3E B3             JSR     MALLOC
AD23 F0 03                BEQ     *+5
AD25 A2 03                LDX     #3
AD27 60                   RTS
AD28 A5 88                LDA     MEMA
AD2A 85 B4                STA     PTR
AD2C A5 89                LDA     MEMA+1
AD2E 85 B5                STA     PTR+1
AD30 20 22 AF             JSR     S1TDS1
AD33 AE 35 08             LDX     CURPLR
AD36 20 D9 AD             JSR     SAVPLR
AD39 AD 0D 08             LDA     OBJCTR
AD3C 20 C9 AD             JSR     SAVCRD
                        ; GET OBJLST MEMORY
AD3F 20 FE AE             JSR     LS1TS1
AD42 A9 02                LDA     #OELELN
AD44 85 8A                STA     MEMB
AD46 A9 00                LDA     #0
AD48 85 8B                STA     MEMB+1
AD4A A5 84                LDA     S2L
AD4C 85 88                STA     MEMA
AD4E A5 85                LDA     S2L+1
AD50 85 89                STA     MEMA+1
AD52 20 3E B3             JSR     MALLOC
AD55 F0 03                BEQ     *+5
AD57 A2 03                LDX     #3
AD59 60                   RTS
                        ; WRITE OBJLST PTR
AD5A 20 0A AF             JSR     S1TLS1
AD5D EE 40 0C             INC     OLSTLN
AD60 A0 01                LDY     #1
AD62 B9 B4 00     LOOP96  LDA     PTR,Y
AD65 91 88                STA     (MEMA),Y
AD67 88                   DEY
AD68 10 F8                BPL     LOOP96
AD6A A2 00                LDX     #0
AD6C 60                   RTS
AD6D A9 FF        ZERPLR  LDA     #$FF
AD6F AE 35 08             LDX     CURPLR
AD72 95 BA                STA     CURFRM+PO,X
AD74 95 BE                STA     CURFRM+XO,X
AD76 95 C2                STA     CURFRM+YO,X
AD78 A2 00                LDX     #0
```

```
AD7A 60                 RTS
                ; SUBROUTINE FIL
                ; REQUIRES FILVAL,MBC, AND MDP
AD7B A5 90      FIL     LDA     MBC
AD7D AA                 TAX
AD7E 05 91              ORA     MBC+1
AD80 F0 1C              BEQ     FIL090
AD82 A0 00              LDY     #0
AD84 AD 67 08   FIL010  LDA     FILVAL
AD87 91 8E              STA     (MDP),Y
AD89 C8                 INY
AD8A D0 02              BNE     FIL020
AD8C E6 8F              INC     MDP+1
AD8E CA         FIL020  DEX
AD8F D0 04              BNE     FIL030
AD91 A5 91              LDA     MBC+1
AD93 F0 09              BEQ     FIL090
AD95 E0 FF      FIL030  CPX     #$FF
AD97 D0 EB              BNE     FIL010
AD99 C6 91              DEC     MBC+1
AD9B 4C 84 AD           JMP     FIL010
AD9E AE 35 08   FIL090  LDX     CURPLR
ADA1 60                 RTS
                ; SUBROUTINE BACACC
                ; REQUIRES Y = INDEX INTO LIST
ADA2 98         BACACC  TYA
ADA3 18                 CLC
ADA4 69 01              ADC     #1
ADA6 85 92              STA     ACC
ADA8 A8                 TAY
ADA9 C0 08              CPY     #MAXBAC
ADAB 90 03              BCC     *+5
ADAD A2 05              LDX     #5      OUT OF LIST RANGE
ADAF 60                 RTS
ADB0 A9 00              LDA     #0
ADB2 85 93              STA     ACC+1
ADB4 20 F9 AE           JSR     DMPY2
ADB7 A9 2F              LDA     #BACLST
ADB9 85 B4              STA     PTR
ADBB A9 0C              LDA     #BACLST/256
ADBD 85 B5              STA     PTR+1
ADBF A2 34              LDX     #PTR-DTAB
ADC1 A0 12              LDY     #ACC-DTAB
ADC3 20 F4 B2           JSR     DADDI
ADC6 A2 00              LDX     #0
ADC8 60                 RTS
                ; SAVE OBJCTR TO CURFRM
ADC9 AE 35 08   SAVCRD  LDX     CURPLR
ADCC 95 BA              STA     CURFRM+P0,X
                ; SAVE X AND Y LOCATIONS TO CURFRM
ADCE BD 11 08           LDA     HORP0,X
ADD1 95 BE              STA     CURFRM+X0,X
ADD3 BD 21 08           LDA     VPOSP0,X
ADD6 95 C2              STA     CURFRM+Y0,X
ADD8 60                 RTS
                ; SAVE DATA
ADD9 8A         SAVPLR  TXA
ADDA 0A                 ASL     A
ADDB A8                 TAY
ADDC B9 ED AC           LDA     PLYRTB+1,Y
```

```
ADDF 85 8D                  STA     MSP+1
ADE1 B9 EC AC               LDA     PLYRTB,Y
ADE4 18                     CLC
ADE5 7D 21 08               ADC     VPOSPO,X
ADE8 85 8C                  STA     MSP
ADEA 90 02                  BCC     *+4
ADEC E6 8D                  INC     MSP+1
ADEE A5 89                  LDA     MEMA+1
ADF0 85 8F                  STA     MDP+1
ADF2 A5 88                  LDA     MEMA
ADF4 18                     CLC
ADF5 69 04                  ADC     #DTSTRT
ADF7 85 8E                  STA     MDP
ADF9 90 02                  BCC     *+4
ADFB E6 8F                  INC     MDP+1
ADFD BD 29 08               LDA     HEIGHT,X
AE00 85 90                  STA     MBC
AE02 A9 00                  LDA     #0
AE04 85 91                  STA     MBC+1
AE06 20 5F B4               JSR     MOVIA
                    ; SAVE DATA LENGTH IN BYTES TO PLAYER DATA
AE09 AE 35 08               LDX     CURPLR
AE0C A0 03                  LDY     #DATLEN
AE0E BD 29 08               LDA     HEIGHT,X
AE11 91 88                  STA     (MEMA),Y
AE13 60                     RTS
                    ; SUBROUTINE
                    ; DO VERT SIZE ROUTINE
                    ; REQUIRES CURPLR
AE14 AD 35 08       VRTSIZ  LDA     CURPLR
AE17 0A                     ASL     A
AE18 A8                     TAY
AE19 B9 EC AC               LDA     PLYRTB,Y
AE1C 85 8C                  STA     MSP
AE1E B9 ED AC               LDA     PLYRTB+1,Y
AE21 85 8D                  STA     MSP+1
                    ;
AE23 A9 4F                  LDA     #PBTEND-1
AE25 8D 0A 08               STA     PTOP
AE28 A9 00                  LDA     #0
AE2A 8D 0B 08               STA     PBOT
AE2D A0 00                  LDY     #0
                    ;
AE2F B1 8C          VSIZLP  LDA     (MSP),Y
AE31 F0 10                  BEQ     TVSEND
AE33 CC 0A 08               CPY     PTOP
AE36 B0 03                  BCS     *+5
AE38 8C 0A 08               STY     PTOP
AE3B CC 0B 08               CPY     PBOT
AE3E 90 03                  BCC     *+5
AE40 8C 0B 08               STY     PBOT
AE43 C8             TVSEND  INY
AE44 C0 50                  CPY     #PBTEND
AE46 90 E7                  BCC     VSIZLP
                    ;
AE48 AD 0A 08               LDA     PTOP
AE4B C9 50                  CMP     #PBTEND
AE4D F0 1C                  BEQ     NOHGHT
AE4F AD 0B 08               LDA     PBOT
AE52 38                     SEC
```

```
AE53 ED 0A 08              SBC     PTOP
AE56 18                    CLC
AE57 69 01                 ADC     #1
AE59 AE 35 08              LDX     CURPLR
AE5C 9D 29 08              STA     HEIGHT,X
AE5F AD 0A 08              LDA     PTOP
AE62 9D 21 08              STA     VPOSPO,X
AE65 9D 19 08              STA     OLDVPO,X
AE68 A2 00                 LDX     #0
AE6A 60                    RTS
AE6B AE 35 08    NOHGHT    LDX     CURPLR
AE6E A9 00                 LDA     #0
AE70 9D 29 08              STA     HEIGHT,X
AE73 A2 00                 LDX     #0
AE75 60                    RTS
                 ; SUBROUTINE
AE76 AD 15 08    INKPOS    LDA     HORMO
AE79 38                    SEC
AE7A FD 11 08              SBC     HORPO,X
AE7D 49 FF                 EOR     #$FF
AE7F 29 07                 AND     #7
AE81 9D 31 08              STA     BITOFF,X
AE84 C9 08                 CMP     #8
AE86 B0 14                 BCS     OUTRNG  OUT OF CURRENT PLAYER RANGE
AE88 AD 25 08              LDA     VPOSMO
AE8B 9D 21 08              STA     VPOSPO,X
AE8E AC 07 08              LDY     CLSLCT
AE91 D0 06                 BNE     WRTONE
AE93 20 A6 B0              JSR     PZERPX
AE96 4C 9C AE              JMP     OUTRNG
AE99 20 84 B0    WRTONE    JSR     PWRTPX
AE9C 60          OUTRNG    RTS
                 ; SUBROUTINE
AE9D AD 4B 08    WFVBFG    LDA     VBFLAG
AEA0 F0 FB                 BEQ     WFVBFG
AEA2 A9 00                 LDA     #0
AEA4 8D 4B 08              STA     VBFLAG
AEA7 60                    RTS
                 ; SUBROUTINE CPLRBD
                 ; REQUIRES X AND HORMO SET
AEA8 AD 15 08    CPLRBD    LDA     HORMO
AEAB 9D 11 08              STA     HORPO,X SET PO LOCATION
AEAE 8D 37 08              STA     PLBND
AEB1 8D 58 08              STA     LB
AEB4 18                    CLC
AEB5 69 07                 ADC     #7
AEB7 8D 38 08              STA     PRBND
AEBA 8D 5A 08              STA     RB
AEBD BD 11 08              LDA     HORPO,X
AEC0 18                    CLC
AEC1 69 30                 ADC     #48     LEFT EDGE OFFSET
AEC3 9D 00 D0              STA     HPOSPO,X
AEC6 A9 4F                 LDA     #PBTEND-1
AEC8 8D 5B 08              STA     BB
AECB 60                    RTS
                 ; SUBROUTINE MESAGE
                 ; REQUIRES PTR
AECC A2 00       MESAGE    LDX     #0      EDITOR
AECE A9 09                 LDA     #PUTREC
AED0 9D 42 03              STA     ICCOM,X
```

```
AED3 B9 E5 A1           LDA     MSGTBL,Y
AED6 9D 44 03           STA     ICBAL,X
AED9 B9 E6 A1           LDA     MSGTBL+1,Y
AEDC 9D 45 03           STA     ICBAH,X
AEDF A9 26              LDA     #38     MAX LENGTH
AEE1 9D 48 03           STA     ICBLL,X
AEE4 A9 00              LDA     #0
AEE6 9D 49 03           STA     ICBLH,X
AEE9 20 56 E4           JSR     CIOV
                ; TEST STATUS??????????????
AEEC 60                 RTS
                ; SUBROUTINE TO MPY 2-BYTE ACC BY POWERS OF 2.
AEED 06 92      DMPY16  ASL     ACC
AEEF 26 93              ROL     ACC+1
AEF1 06 92      DMPY8   ASL     ACC
AEF3 26 93              ROL     ACC+1
AEF5 06 92      DMPY4   ASL     ACC
AEF7 26 93              ROL     ACC+1
AEF9 06 92      DMPY2   ASL     ACC
AEFB 26 93              ROL     ACC+1
AEFD 60                 RTS
                ; SUBROUTINE
AEFE A0 07      LS1TS1  LDY     #7
AF00 B9 94 00   LS1LP   LDA     LS1L,Y
AF03 99 80 00           STA     S1L,Y
AF06 88                 DEY
AF07 10 F7              BPL     LS1LP
AF09 60                 RTS
                ; SUBROUTINE
AF0A A0 07      S1TLS1  LDY     #7
AF0C B9 80 00   S1TLLP  LDA     S1L,Y
AF0F 99 94 00           STA     LS1L,Y
AF12 88                 DEY
AF13 10 F7              BPL     S1TLLP
AF15 60                 RTS
                ; SUBROUTINE
AF16 A0 07      DS1TS1  LDY     #7
AF18 B9 9C 00   DS1LP   LDA     DS1L,Y
AF1B 99 80 00           STA     S1L,Y
AF1E 88                 DEY
AF1F 10 F7              BPL     DS1LP
AF21 60                 RTS
                ; SUBROUTINE
AF22 A0 07      S1TDS1  LDY     #7
AF24 B9 80 00   S1TDLP  LDA     S1L,Y
AF27 99 9C 00           STA     DS1L,Y
AF2A 88                 DEY
AF2B 10 F7              BPL     S1TDLP
AF2D 60                 RTS
                ; SUBROUTINE
AF2E 84 92      OBJACC  STY     ACC
AF30 A9 00              LDA     #0
AF32 85 93              STA     ACC+1
AF34 20 F9 AE           JSR     DMPY2
AF37 A9 2D              LDA     #OBJLST-1
AF39 85 B4              STA     PTR
AF3B A9 0C              LDA     #OBJLST-1/256
AF3D 85 B5              STA     PTR+1
AF3F A2 34              LDX     #PTR-DTAB
AF41 A0 12              LDY     #ACC-DTAB
```

```
AF43 20 04 B3              JSR     DSUBI
AF46 A2 00                 LDX     #0
AF48 60                    RTS
                    ; SUBROUTINE
AF49 84 92          FRMACC STY     ACC
AF4B A9 00                 LDA     #0
AF4D 85 93                 STA     ACC+1
AF4F 20 ED AE              JSR     DMPY16
AF52 A9 6F                 LDA     #FRMLST
AF54 85 B4                 STA     PTR
AF56 A9 0B                 LDA     #FRMLST/256
AF58 85 B5                 STA     PTR+1
AF5A A2 34                 LDX     #PTR-DTAB
AF5C A0 12                 LDY     #ACC-DTAB
AF5E 20 F4 B2              JSR     DADDI
AF61 A2 00                 LDX     #0
AF63 60                    RTS
                    ; SUBROUTINE TO SET CURSOR SCREEN BOUNDS
AF64 A9 00          CSCNBD LDA     #0
AF66 8D 58 0B              STA     LB
AF69 8D 59 0B              STA     TB
AF6C A9 9F                 LDA     #PRTEND-1
AF6E 8D 5A 0B              STA     RB
AF71 A9 4F                 LDA     #PBTEND-1
AF73 8D 5B 0B              STA     BB
AF76 60                    RTS
                    ; SUBROUTINE BLDSCN
AF77 20 8B AF       BLDSCN JSR     BKPLOP
                    ; POSITION PLAYERS ON SCREEN
                    ; PLOP OUT PLAYERS 0-3
AF7A A2 03                 LDX     #3
AF7C 8E 35 0B              STX     CURPLR
                    ; ????????????????????????????
AF7F AE 35 0B       PPLPLP LDX     CURPLR
AF82 20 B7 AF              JSR     PPLOP
AF85 CE 35 0B              DEC     CURPLR
AF88 10 F5                 BPL     PPLPLP
AF8A 60                    RTS
                    ; SUBROUTINE TO PLOP OUT BACCKGROUND
AF8B A0 03          BKPLOP LDY     #BAKGND         =2
AF8D B9 B6 00              LDA     CURFRM,Y
AF90 A8                    TAY
AF91 20 A2 AD              JSR     BACACC
AF94 A0 00                 LDY     #0
AF96 B1 B4                 LDA     (PTR),Y
AF98 AA                    TAX
AF99 C8                    INY
AF9A B1 B4                 LDA     (PTR),Y
AF9C A0 04                 LDY     #4
AF9E 8D 69 0B              STA     TMP2
AFA1 8A                    TXA
AFA2 18                    CLC
AFA3 69 02                 ADC     #2      BDATA HEADER OFFSET
AFA5 91 AE                 STA     (DLPTR),Y
AFA7 85 AC                 STA     RAMPTR
AFA9 90 03                 BCC     *+5
AFAB EE 69 0B              INC     TMP2
AFAE AD 69 0B              LDA     TMP2
AFB1 C8                    INY
AFB2 85 AD                 STA     RAMPTR+1
```

```
AFB4 91 AE              STA     (DLPTR),Y
AFB6 60         ENDPLP  RTS
                ; SUBROUTINE PPLOP
AFB7 B5 BA      PPLOP   LDA     CURFRM+PO,X     GET OBJLST NO.
AFB9 C9 FF              CMP     #$FF
AFBB D0 0F              BNE     PLPLYR
                ; POSITION PLAYER OFF SCREEN IF NO ENTRY
AFBD A9 A8              LDA     #PRTEND+8
AFBF AE 35 08           LDX     CURPLR
AFC2 9D 11 08           STA     HORPO,X
AFC5 18                 CLC
AFC6 69 30              ADC     #48
AFC8 9D 00 D0           STA     HPOSPO,X
AFCB 60                 RTS

                ;
AFCC A8         PLPLYR  TAY
AFCD 20 2E AF           JSR     OBJACC
AFD0 20 E2 AF           JSR     WRITE
                ; SET HORIZ POSITION
AFD3 AE 35 08           LDX     CURPLR
AFD6 B5 BE              LDA     CURFRM+XO,X     X LOC.

AFD8 9D 11 08           STA     HORPO,X
AFDB 18                 CLC
AFDC 69 30              ADC     #48
AFDE 9D 00 D0           STA     HPOSPO,X
                ;
AFE1 60         ENDPP   RTS
                ; GET HEIGHT FROM PLAYER DATA HEADER
AFE2 A0 00      WRITE   LDY     #0
AFE4 B1 B4              LDA     (PTR),Y
AFE6 AA                 TAX
AFE7 C8                 INY
AFE8 B1 B4              LDA     (PTR),Y
AFEA 85 8D              STA     MSP+1
AFEC 86 8C              STX     MSP
AFEE A0 03              LDY     #DATLEN
AFF0 B1 8C              LDA     (MSP),Y
AFF2 AE 35 08           LDX     CURPLR
AFF5 9D 29 08           STA     HEIGHT,X
                ; UPDATE OBJPTR TO START OF ACTUAL DATA
AFF8 A5 8C              LDA     MSP
AFFA 18                 CLC
AFFB 69 04              ADC     #DTSTRT
AFFD 85 8C              STA     MSP
AFFF 90 02              BCC     *+4
B001 E6 8D              INC     MSP+1
                ; SET UP DEST. POINTER
B003 AD 35 08           LDA     CURPLR
B006 0A                 ASL     A
B007 A8                 TAY
B008 B9 EC AC           LDA     PLYRTB,Y
B00B 8D 68 08           STA     TMP1
B00E B9 ED AC           LDA     PLYRTB+1,Y
B011 8D 69 08           STA     TMP2
                ; ERASE OLD PLAYER
B014 85 8F              STA     MDP+1
B016 AE 35 08           LDX     CURPLR
B019 AD 68 08           LDA     TMP1
```

```
B01C 18                 CLC
B01D 7D 19 08           ADC     OLDVPO,X
B020 85 8E              STA     MDP
B022 90 02              BCC     *+4
B024 E6 8F              INC     MDP+1
                ;
B026 BD 2D 08           LDA     OLDHGT,X
B029 85 90              STA     MBC
B02B A9 00              LDA     #0
B02D 85 91              STA     MBC+1
B02F 8D 67 08           STA     FILVAL
B032 20 7B AD           JSR     FIL
                ;
B035 B5 C2              LDA     CURFRM+YO,X     Y LOC.
B037 9D 21 08           STA     VPOSPO,X
B03A 9D 19 08           STA     OLDVPO,X
                ; WRITE NEW PLAYER
B03D AD 69 08           LDA     TMP2
B040 85 8F              STA     MDP+1
B042 AD 68 08           LDA     TMP1
B045 18                 CLC
B046 7D 21 08           ADC     VPOSPO,X        ???????????
B049 85 8E              STA     MDP
B04B 90 02              BCC     *+4
B04D E6 8F              INC     MDP+1
                ;
B04F BD 29 08           LDA     HEIGHT,X
B052 9D 2D 08           STA     OLDHGT,X
B055 85 90              STA     MBC
B057 A9 00              LDA     #0
B059 85 91              STA     MBC+1
                ; WRITE THE DATA
B05B 20 5F B4           JSR     MOVIA
B05E 60                 RTS
                ;
                ;
                ; SUBROUTINE PUT1CH
                ; REQUIRES Y = CHAR CODE, X = 16(IOCB#)
B05F A9 0B      PUT1CH  LDA     #PUTCHR
B061 9D 42 03           STA     ICCOM,X
B064 A9 00              LDA     #0
B066 9D 48 03           STA     ICBLL,X
B069 9D 49 03           STA     ICBLH,X
B06C 98                 TYA
B06D 20 56 E4           JSR     CIOV
B070 60                 RTS
                ; SUBROUTINE TO GET KEY FROM KEYBOARD
                ;
B071 A2 10      GETKEE  LDX     #$10    KEYBOARD
B073 A9 07              LDA     #GETCHR
B075 9D 42 03           STA     ICCOM,X
B078 A9 00              LDA     #0      BUFFER LENGTH OF 0
B07A 9D 48 03           STA     ICBLL,X
B07D 9D 49 03           STA     ICBLH,X
B080 20 56 E4           JSR     CIOV
B083 60                 RTS
                ; SUBROUTINE PWRTPX
                ; REQUIRES X, VPOSPO, AND BITOFF FOR PLAYER X
                ; WRITE A PIXEL TO PLAYER X.
B084 8A         PWRTPX  TXA
```

```
B085 0A          ASL    A
B086 A8          TAY
B087 B9 ED AC    LDA    PLYRTB+1,Y
B08A 85 8F       STA    MDP+1
B08C B9 EC AC    LDA    PLYRTB,Y
B08F 85 8E       STA    MDP
B091 BC 21 08    LDY    VPOSPO,X
B094 BD 31 08    LDA    BITOFF,X
B097 AA          TAX
B098 B1 8E       LDA    (MDP),Y
B09A 3D DC AC    AND    ZERTBL,X
B09D 1D E4 AC    ORA    ONETBL,X
B0A0 91 8E       STA    (MDP),Y
B0A2 AE 35 08    LDX    CURPLR
B0A5 60          RTS
;
B0A6 8A          PZERPX TXA
B0A7 0A          ASL    A
B0A8 A8          TAY
B0A9 B9 ED AC    LDA    PLYRTB+1,Y
B0AC 85 8F       STA    MDP+1
B0AE B9 EC AC    LDA    PLYRTB,Y
B0B1 85 8E       STA    MDP
B0B3 BC 21 08    LDY    VPOSPO,X
B0B6 BD 31 08    LDA    BITOFF,X
B0B9 AA          TAX
B0BA B1 8E       LDA    (MDP),Y
B0BC 3D DC AC    AND    ZERTBL,X
B0BF 91 8E       STA    (MDP),Y
B0C1 AE 35 08    LDX    CURPLR
B0C4 60          RTS
                 ;
                 ;
                 ; SUBROUTINE FILSCN
                 ;
                 ; REQUIRES COLOUR AND RAMPTR TO BE SET.
                 ; ALSO VALUES FOR SCNEND AND RITEND.
                 ; USES 2-BYTE POINTER PTR.
                 ;
B0C5 A2 27       FILSCN LDX    #SCNEND-1
B0C7 8A          CLRLUP TXA
B0C8 0A                 ASL    A
B0C9 A8                 TAY
B0CA A5 AD              LDA    RAMPTR+1
B0CC 18                 CLC
B0CD 79 ED B0           ADC    LPTRTB+1,Y
B0D0 85 B5              STA    PTR+1
B0D2 A5 AC              LDA    RAMPTR
B0D4 18                 CLC
B0D5 79 EC B0           ADC    LPTRTB,Y
B0D8 85 B4              STA    PTR
B0DA 90 02              BCC    *+4
B0DC E6 B5              INC    PTR+1
                 ;
B0DE A0 13              LDY    #RITEND/4-1
B0E0 AD 66 08           LDA    COLOUR
B0E3 91 B4       CLOOP2 STA    (PTR),Y
B0E5 88                 DEY
B0E6 10 FB              BPL    CLOOP2
B0E8 CA                 DEX
```

```
BOE9 10 DC                  BPL     CLRLUP
BOEB 60                     RTS
                            ;
                            ;
                            ;
```

```
0014                    SCRWID  =20
                        ;
B0EC 00 00              LPTRTB  .WORD   0*SCRWID
B0EE 14 00                      .WORD   1*SCRWID
B0F0 28 00                      .WORD   2*SCRWID
B0F2 3C 00                      .WORD   3*SCRWID
B0F4 50 00                      .WORD   4*SCRWID
B0F6 64 00                      .WORD   5*SCRWID
B0F8 78 00                      .WORD   6*SCRWID
B0FA 8C 00                      .WORD   7*SCRWID
B0FC A0 00                      .WORD   8*SCRWID
B0FE B4 00                      .WORD   9*SCRWID
B100 C8 00                      .WORD   10*SCRWID
B102 DC 00                      .WORD   11*SCRWID
B104 F0 00                      .WORD   12*SCRWID
B106 04 01                      .WORD   13*SCRWID
B108 18 01                      .WORD   14*SCRWID
B10A 2C 01                      .WORD   15*SCRWID
B10C 40 01                      .WORD   16*SCRWID
B10E 54 01                      .WORD   17*SCRWID
B110 68 01                      .WORD   18*SCRWID
B112 7C 01                      .WORD   19*SCRWID
B114 90 01                      .WORD   20*SCRWID
B116 A4 01                      .WORD   21*SCRWID
B118 B8 01                      .WORD   22*SCRWID
B11A CC 01                      .WORD   23*SCRWID
B11C E0 01                      .WORD   24*SCRWID
B11E F4 01                      .WORD   25*SCRWID
B120 08 02                      .WORD   26*SCRWID
B122 1C 02                      .WORD   27*SCRWID
B124 30 02                      .WORD   28*SCRWID
B126 44 02                      .WORD   29*SCRWID
B128 58 02                      .WORD   30*SCRWID
B12A 6C 02                      .WORD   31*SCRWID
B12C 80 02                      .WORD   32*SCRWID
B12E 94 02                      .WORD   33*SCRWID
B130 A8 02                      .WORD   34*SCRWID
B132 BC 02                      .WORD   35*SCRWID
B134 D0 02                      .WORD   36*SCRWID
B136 E4 02                      .WORD   37*SCRWID
B138 F8 02                      .WORD   38*SCRWID
B13A 0C 03                      .WORD   39*SCRWID
B13C 20 03                      .WORD   40*SCRWID
B13E 34 03                      .WORD   41*SCRWID
B140 48 03                      .WORD   42*SCRWID
B142 5C 03                      .WORD   43*SCRWID
B144 70 03                      .WORD   44*SCRWID
B146 84 03                      .WORD   45*SCRWID
B148 98 03                      .WORD   46*SCRWID
B14A AC 03                      .WORD   47*SCRWID
```

```
                        ; READ PIXEL AND WRITE PIXEL SUBROUTINES.

                        ;SUBROUTINE READPX READS THE COLOR OF THE SCREEN PIXEL AT THE COORDINATES
                        ;PASSED AS ARGUMENTS.
                        ;       ARGS --
                        ;X IS THE HORIZONTAL COORDINATE (IN PIXELS) OF THE PIXEL TO BE READ.
                        ;Y IS THE VERTICAL COORDINATE (IN PIXELS) OF THE PIXEL TO BE READ.
                        ;       SCREEN ORIGIN IS AT UPPER LEFT.
                        ;A IS RETURNED AS THE COLOR (0, 1, 2, OR 3) OF THE PIXEL IN QUESTION.

                        ;READPX USES THE GLOBAL TABLE LPTRTB (ADRESSES OF SCREEN LINES).
                        ;READPX TAKES ABOUT 100 CYCLES PER CALL.

                        ;SUBROUTINE WRITPX WRITES A GIVEN COLOR TO A PIXEL AT GIVEN SCREEN
                        ;COORDINATES.
                        ;       ARGS --
                        ;X IS THE HORIZONTAL COORDINATE (IN PIXELS) OF THE PIXEL TO BE WRITTEN.
                        ;Y IS THE VERTICAL COORDINATE (IN PIXELS) OF THE PIXEL TO BE WRITTEN.
                        ;       SCREEN ORIGIN IS AT UPPER LEFT.
                        ;A IS THE COLOR TO BE WRITTEN TO THE PIXEL (0, 1, 2, OR 3).

                        ;WRITPX USES THE GLOBAL TABLE LPTRTB.
                        ;WRITPX TAKES ABOUT 100 CYCLES PER CALL.

                        ;VARIABLES.
B14C                            P=*
                                *=V
086A                    COLR    *=*+1   COLOR (0,1,2, OR 3) TO BE WRITTEN
086B                    PAKCOL  *=*+1   4 OF THE SAME 2-BIT COLOR PATTERN PACKED INTO A BYTE
086C                    WRTEMP  *=*+1   TEMPORARY
086D                            V=*
                                *=P
B14C 98                 READPX  TYA
B14D 0A                         ASL     A
B14E A8                         TAY
B14F A5 AD                      LDA     RAMPTR+1
B151 18                         CLC
B152 79 ED B0                   ADC     LPTRTB+1,Y
B155 85 B5                      STA     PTR+1
B157 B9 EC B0                   LDA     LPTRTB,Y
B15A 18                         CLC
B15B 65 AC                      ADC     RAMPTR
B15D 85 B4                      STA     PTR
B15F 90 02                      BCC     *+4
B161 E6 B5                      INC     PTR+1

B163 8A                         TXA
B164 4A                         LSR     A
B165 4A                         LSR     A
B166 A8                         TAY                     BYTE INDEX IN LINE
B167 8A                         TXA
B168 29 03                      AND     #3
B16A AA                         TAX                     BIT INDEX
B16B B1 B4                      LDA     (PTR),Y
B16D 3D C6 B1                   AND     PX11TB,X        EXTRACT PIXEL

B170 8D 6C 08                   STA     WRTEMP
                        ;RIGHT JUSTIFY PIXEL IN A.
B173 8A                         TXA
B174 49 03                      EOR     #3
```

```
B176 A8                     TAY
B177 88                     DEY
B178 30 09                  BMI     GRJUS
B17A 4E 6C 08       RJUSLP  LSR     WRTEMP
B17D 4E 6C 08               LSR     WRTEMP
B180 88                     DEY
B181 10 F7                  BPL     RJUSLP
B183                GRJUS
B183 AD 6C 08               LDA     WRTEMP
B186 60                     RTS             END OF READPX

B187 8D 6A 08       WRITPX  STA     COLR
B18A 98                     TYA
B18B 0A                     ASL     A
B18C A8                     TAY
B18D A5 AD                  LDA     RAMPTR+1
B18F 18                     CLC
B190 79 ED B0               ADC     LPTRTB+1,Y
B193 85 B5                  STA     PTR+1
B195 B9 EC B0               LDA     LPTRTB,Y
B198 18                     CLC
B199 65 AC                  ADC     RAMPTR
B19B 85 B4                  STA     PTR
B19D 90 02                  BCC     *+4
B19F E6 B5                  INC     PTR+1

B1A1 AC 6A 08               LDY     COLR
B1A4 B9 CE B1               LDA     CPATTB,Y
B1A7 8D 6B 08               STA     PAKCOL          PACK COLOR INTO THIS BYTE 4 TIMES

B1AA 8A                     TXA
B1AB 4A                     LSR     A
B1AC 4A                     LSR     A
B1AD A8                     TAY                     BYTE INDEX IN LINE
B1AE 8A                     TXA
B1AF 29 03                  AND     #3
B1B1 AA                     TAX                     BIT INDEX

B1B2 B1 B4                  LDA     (PTR),Y
B1B4 3D CA B1               AND     PX00TB,X
B1B7 8D 6C 08               STA     WRTEMP          CLEAR PIXEL
B1BA AD 6B 08               LDA     PAKCOL
B1BD 3D C6 B1               AND     PX11TB,X
B1C0 0D 6C 08               ORA     WRTEMP          OR IN NEW COLOR
B1C3 91 B4                  STA     (PTR),Y
B1C5 60                     RTS             END OF WRITPX

B1C6 C0 30 0C       PX11TB  .BYTE   $C0,$30,$0C,$03
B1C9 03
B1CA 3F CF F3       PX00TB  .BYTE   $3F,$CF,$F3,$FC
B1CD FC
B1CE 00 55 AA       CPATTB  .BYTE   $00,$55,$AA,$FF
B1D1 FF
```

```
                        ; CURSOR MOVEMENT SUBROUTINE.

                        ;SUBROUTINE CURSOR MOVES THE CURSOR (MO) ABOUT ON THE SCREEN ACCORDING
                        ;TO THE CONTROL OF THE JOYSTICK (#0).  IT CLEARS THE OLD CURSOR GRAPHICS,
                        ;UPDATES THE CURSOR HORIZONTAL AND VERTICAL COORDINATES FROM THE
                        ;JOYSTICK INPUT, AND THEN WRITES THE CURSOR GRAPHICS AT THEIR NEW
                        ;POSITION.

                        ;
                        ;
                        ; X = STICK NO.
                        ;
                        ; GLOBALS - VDELTA,HDELTA,HORMO,VPOSMO,LB,TB,RB,BB
B1D2                            P=*
                                *=V
086D                    PRVPTA  *=*+1   PREVIOUS FRAME VALUE OF PORTA
086E                    RPTCNT  *=*+1   FRAME COUNT FOR JOYSTICK MOVEMENT AUTO-REPEAT
086F                            V=*
                                *=P

                        ;DECIDE WHETHER TO MOVE THE CURSOR THIS FRAME BASED ON WHETHER A
                        ;JOYSTICK SWITCH WAS JUST CLOSED, AND THE AUTO-REPEAT FRAME COUNT.
B1D2 AD 6D 08           CURSOR  LDA     PRVPTA  INITIALIZED???????????
B1D5 29 0F                      AND     #$0F
B1D7 AC 78 02                   LDY     STICK0
B1DA 8C 6D 08                   STY     PRVPTA
B1DD C9 0F                      CMP     #$0F
B1DF D0 11                      BNE     NOEDGE
                        ; IF PRVPTR IS CLOSED, READ STICK
B1E1 AD 78 02                   LDA     STICK0
B1E4 29 0F                      AND     #$0F
B1E6 C9 0F                      CMP     #$0F
B1E8 F0 08                      BEQ     NOEDGE
                        ; COME HERE IF STICK NOT OPEN
B1EA A9 0F                      LDA     #15     FRAMES AFTER EDGE TILL 1ST AUTO-REPEAT
B1EC 8D 6E 08                   STA     RPTCNT
B1EF 4C 05 B2                   JMP     ACTION

B1F2 AD 6E 08           NOEDGE  LDA     RPTCNT
B1F5 F0 0D                      BEQ     NOACTN
B1F7 CE 6E 08                   DEC     RPTCNT
B1FA D0 08                      BNE     NOACTN
B1FC A9 02                      LDA     #2      FRAMES BETWEEN AUTO-REPEATS
B1FE 8D 6E 08                   STA     RPTCNT
B201 4C 05 B2                   JMP     ACTION

B204 60                 NOACTN  RTS

                        ;ERASE OLD CURSOR.
B205 AE 1D 08           ACTION  LDX     OLDVMO
B208 20 C5 B2                   JSR     ERSMSL
                        ;
B20B A0 04                      LDY     #4      MO
B20D 20 1E B2                   JSR     JOYRD
B210 AD 25 08                   LDA     VPOSMO
B213 8D 1D 08                   STA     OLDVMO
B216 AA                         TAX
B217 AD 15 08                   LDA     HORMO
B21A 20 B6 B2                   JSR     WRTMSL
B21D 60                         RTS             END OF CURSOR
```

```
                        ; SUBROUTINES
                        ; READ JOYSTICK AND UPDATE HORPO AND VPOSPO FOR PLAYERS.
                        ; X IS STICK NO.
                        ; Y IS PLAYER NO.
                        ; BOUNDS TB,BB,RB,LB,HWID,VWID AND SPEEDS VDELTA AND HDELTA MUST BE SET.
                        ;
B21E AD 5A 08           JOYRD   LDA     RB
B221 38                         SEC
B222 ED 58 08                   SBC     LB
B225 18                         CLC
B226 69 01                      ADC     #1
B228 8D 08 08                   STA     HWID
                        ;
B22B AD 5B 08                   LDA     BB
B22E 38                         SEC
B22F ED 59 08                   SBC     TB
B232 18                         CLC
B233 69 01                      ADC     #1
B235 8D 09 08                   STA     VWID
B238 AD 78 02                   LDA     STICKO
B23B 8D 68 08                   STA     TMP1
B23E 29 0F                      AND     #$0F
B240 C9 0F                      CMP     #$0F
B242 F0 71                      BEQ     JDON
B244 A2 00                      LDX     #0
B246 86 4D                      STX     ATRACT
B248 29 01                      AND     #1
B24A D0 18                      BNE     CHKDWN
B24C B9 21 08                   LDA     VPOSPO,Y
B24F 38                         SEC
B250 F9 43 08                   SBC     VDELTA,Y
B253 AA                         TAX
B254 38                         SEC
B255 ED 59 08                   SBC     TB
B258 CD 09 08                   CMP     VWID
B25B 90 03                      BCC     *+5
B25D AE 59 08                   LDX     TB
B260 8A                         TXA
B261 99 21 08                   STA     VPOSPO,Y

B264                    CHKDWN
B264 AD 68 08                   LDA     TMP1
B267 29 02                      AND     #2
B269 D0 12                      BNE     CHKLFT
B26B B9 21 08                   LDA     VPOSPO,Y
B26E 18                         CLC
B26F 79 43 08                   ADC     VDELTA,Y
B272 CD 5B 08                   CMP     BB
B275 90 03                      BCC     *+5
B277 AD 5B 08                   LDA     BB
B27A 99 21 08                   STA     VPOSPO,Y
B27D                    CHKLFT
B27D AD 68 08                   LDA     TMP1
B280 29 04                      AND     #4
B282 D0 18                      BNE     CHKRIT
B284 B9 11 08                   LDA     HORPO,Y
B287 38                         SEC
B288 F9 3B 08                   SBC     HDELTA,Y
B28B AA                         TAX
```

```
B28C 38                 SEC
B28D ED 58 08           SBC     LB
B290 CD 08 08           CMP     HWID
B293 90 03              BCC     *+5
B295 AE 58 08           LDX     LB
B298 8A                 TXA
B299 99 11 08           STA     HORPO,Y
B29C                CHKRIT
B29C AD 68 08           LDA     TMP1
B29F 29 08              AND     #8
B2A1 D0 12              BNE     JDON
B2A3 B9 11 08           LDA     HORPO,Y
B2A6 18                 CLC
B2A7 79 3B 08           ADC     HDELTA,Y
B2AA CD 5A 08           CMP     RB
B2AD 90 03              BCC     *+5
B2AF AD 5A 08           LDA     RB
B2B2 99 11 08           STA     HORPO,Y
B2B5 60             JDON    RTS
                    ;PUT CURSOR AT NEW LOCATION.
                    ; A HAS HORIZ. COORDINATE.
                    ; X HAS VERT. COORDINATE.
                    ; Y HAS MISSILE NO.
B2B6 18             WRTMSL  CLC
B2B7 69 30              ADC     #48             LEFT EDGE ADJUST
B2B9 8D 04 D0           STA     HPOSM0

B2BC BD 90 05           LDA     GXMISL+16,X
B2BF 09 02              ORA     #2
B2C1 9D 90 05           STA     GXMISL+16,X

B2C4 60                 RTS

                    ;
B2C5 BD 90 05       ERSMSL  LDA     GXMISL+16,X
B2C8 29 FC              AND     #$FC
B2CA 9D 90 05           STA     GXMISL+16,X
B2CD 60                 RTS
```

```
                        ;
                        ; DOUBLE PRECISION ROUTINES
                        ;
                        ; ALL VARIABLES ARE ACCESSED VIA THEIR OFFSET FROM SYMBOL 'DTAB'.
                        ; NORMALLY THE X AND/OR Y REGISTERS CONTAIN THE 'DTAB' OFFSET
                        ; VALUES TO THE VARIABLE(S) TO BE DEALT WITH.
                        ;


                        ;
                        ; DCMPI -- DOUBLE BYTE UNSIGNED COMPARE INDEXED
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = DATA #1 OFFSET
                        ;       Y = DATA #2 OFFSET
                        ;
                        ;       JSR     DCMPI
                        ;       BEQ     DTAB(X) = DTAB(Y)
                        ;       BCS     DTAB(X) >= DTAB(Y)
                        ;       BCC     DTAB(X) < DTAB(Y)
                        ;
                        ;       CC = DTAB(X) : DTAB(Y)  (UNSIGNED)
                        ;
B2CE B5 81      DCMPI   LDA     DTAB+1,X        ; COMPARE MSBS.
B2D0 D9 81 00           CMP     DTAB+1,Y
B2D3 D0 05              BNE     DCM090          ; NOT EQUAL -- ALL DONE.

B2D5 B5 80              LDA     DTAB,X          ; EQUAL -- COMPARE LSBS.
B2D7 D9 80 00           CMP     DTAB,Y

B2DA 60         DCM090  RTS


                        ; THE CODE THAT FOLLOWS USES THE FOLLOWING OBSERVED FACT: WHEN THE SUBTRAHEND
                        ; IS LESS THAN THE SUBTRACTOR, THE 'V' BIT AND THE 'N' BIT IN THE CC ARE OF
                        ; OPPOSITE VALUE.

B2DB 10 06      DSC030  BPL     DSC035          ; (N=0).

B2DD 70 06              BVS     DSC037          ; (N=1,V=1).

B2DF A9 01      DSC032  LDA     #1              ; SEE CC FOR "LESS THAN" COMPARE.
B2E1 18                 CLC
B2E2 60                 RTS

B2E3 70 FA      DSC035  BVS     DSC032          ; (N=0,V=1).

B2E5 A9 01      DSC037  LDA     #1              ; SET CC FOR "GREATER THAN" COMPARE.
B2E7 38                 SEC
B2E8 60                 RTS
```

```
                        ;
                        ; DMOVI -- DOUBLE BYTE MOVE INDEXED
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = DESTINATION OFFSET
                        ;       Y = SOURCE OFFSET
                        ;
                        ;       JSR     DMOVI
                        ;
                        ;       DTAB(X) = DTAB(Y)
                        ;
B2E9 B9 80 00       DMOVI   LDA     DTAB,Y
B2EC 95 80                  STA     DTAB,X

B2EE B9 81 00               LDA     DTAB+1,Y
B2F1 95 81                  STA     DTAB+1,X

B2F3 60                     RTS


                        ;
                        ; DADDI -- DOUBLE PRECISION ADD
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET TO
                        ;       Y = OFFSET TO
                        ;
                        ;       JSR     DADDI
                        ;
                        ;       DTAB(X) = DTAB(X) + DTAB(Y)
                        ;
B2F4 B5 80          DADDI   LDA     DTAB,X
B2F6 18                     CLC
B2F7 79 80 00               ADC     DTAB,Y
B2FA 95 80                  STA     DTAB,X

B2FC B5 81                  LDA     DTAB+1,X
B2FE 79 81 00               ADC     DTAB+1,Y
B301 95 81                  STA     DTAB+1,X

B303 60                     RTS


                        ;
                        ; DSUBI -- DOUBLE PRECISION SUBTRACT
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET
                        ;       Y = OFFSET
                        ;
                        ;       JSR     DSUBI
                        ;       BEQ     RESULT = 0
                        ;
                        ;       DTAB(X) = DTAB(X) - DTAB(Y)
                        ;
```

```
B304 B5 80        DSUBI   LDA     DTAB,X
B306 38                   SEC
B307 F9 80 00             SBC     DTAB,Y
B30A 95 80                STA     DTAB,X

B30C B5 81                LDA     DTAB+1,X
B30E F9 81 00             SBC     DTAB+1,Y
B311 95 81                STA     DTAB+1,X

B313 15 80                ORA     DTAB,X          ; SET CC FOR ZERO TEST.

B315 60                   RTS
```

```
                        ;
                        ; ACCUMULATOR FUNCTIONS -- ASSUME THE EXISTENCE OF A DOUBLE PRECISION
                        ; VARIABLE WITHIN 'DTAB' NAMED 'ACC'.
                        ;

                        ;
                        ; DLOADA -- LOAD 'ACC' WITH DATA
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = OFFSET TO SOURCE DATA
                        ;
                        ;       JSR     DLOADA
                        ;
                        ;       X = ACC OFFSET
                        ;       'ACC' = DTAB(Y)
                        ;
B316 A2 12              DLOADA  LDX     #ACC-DTAB
B318 4C E9 B2                   JMP     DMOVI

                        ;
                        ; DSTORA -- STORE 'ACC' TO LOCATION
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       X = OFFSET TO DESTINATION
                        ;
                        ;       JSR     DSTORA
                        ;
                        ;       Y = 'ACC' OFFSET
                        ;       DTAB(X) = 'ACC'
                        ;
B31B A0 12              DSTORA  LDY     #ACC-DTAB
B31D 4C E9 B2                   JMP     DMOVI

                        ;
                        ; DADDA -- ADD DATA TO 'ACC'
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = OFFSET TO DATA
                        ;
                        ;       JSR     DADDA
                        ;
                        ;       X = 'ACC' OFFSET
                        ;       'ACC' = 'ACC' + DTAB(Y)
                        ;
B320 A2 12              DADDA   LDX     #ACC-DTAB
B322 4C F4 B2                   JMP     DADDI

                        ;
                        ; DSUBA -- SUBTRACT DATA FROM 'ACC'
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = OFFSET TO DATA
                        ;
```

```
                        ;       JSR     DSUBA
                        ;       BEQ     RESULT = 0
                        ;
                        ;       X = 'ACC' OFFSET
                        ;       'ACC' = 'ACC' - DTAB(Y)
                        ;
B325 A2 12              DSUBA   LDX     #ACC-DTAB
B327 4C 04 B3                   JMP     DSUBI


                        ;
                        ; DCMPA -- COMPARE 'ACC' WITH DATA (UNSIGNED)
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       Y = DATA OFFSET
                        ;
                        ;       JSR     DCMPA
                        ;
                        ;       CC = 'ACC' : DTAB(Y)  (UNSIGNED)
                        ;       X = 'ACC' OFFSET
                        ;
B32A A2 12              DCMPA   LDX     #ACC-DTAB
B32C 4C CE B2                   JMP     DCMPI
```

```
                        ;
                        ; MEMORY MANAGEMENT PACKAGE
                        ;
                        ; AVAILABLE MEMORY IS DIVIDED INTO TWO REGIONS WHICH GROW TOWARD EACH
                        ; OTHER;  THE REGIONS ARE DEFINED BY FOUR POINTER VARIABLES:
                        ;
                        ;       'S1L' POINTS TO BOTTOM OF REGION #1
                        ;       'S1H' POINTS TO FIRST UNUSED LOCATION ABOVE REGION #1
                        ;       'S2L' POINTS TO BOTTOM OF REGION #2
                        ;       'S2H' POINTS TO FIRST UNUSED LOCATION ABOVE REGION #2
                        ;
                        ; THREE ROUTINES ARE PROVIDED TO ALLOCATE AND DEALLOCATE MEMORY:
                        ;
                        ;       'MINIT' IS USED TO INITIALIZE MEMORY
                        ;       'MALLOC' IS USED TO ALLOCATE MEMORY
                        ;       'MDEALL' IS USED TO DEALLOCATE MEMORY
                        ;
                        ; THE TWO REGIONS ARE MAINTAINED AS TWO COMPRESSED STACKS;  ALLOCATION
                        ; AND DEALLOCATION INVOLVES THE MOVEMENT OF DATA TO CREATE AND
                        ; ELIMINATE HOLES IN THE STACKS.
                        ;

B32F A2 02      MINIT   LDX     #S1H-DTAB       ; S1H = S1L
B331 A0 00              LDY     #S1L-DTAB
B333 20 E9 B2           JSR     DMOVI

B336 A2 04              LDX     #S2L-DTAB       ; S2L = S2H.
B338 A0 06              LDY     #S2H-DTAB
B33A 20 E9 B2           JSR     DMOVI

B33D 60                 RTS


                        ;
                        ; MALLOC -- MEMORY ALLOCATE
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'MEMA' CONTAINS THE ADDRESS OF THE START OF ALLOCATION
                        ;          REGION #1:  DATA AT START ADDRESS AND ABOVE ARE MOVED UP.
                        ;          REGION #2:  DATA BELOW START ADDRESS ARE MOVED DOWN.
                        ;       'MEMB' CONTAINS THE NUMBER OF BYTES TO ALLOCATE
                        ;
                        ;       JSR     MALLOC
                        ;       BNE     NOT ENOUGH MEMORY TO SATISFY ALLOCATION
                        ;
                        ;       'MEMA' CONTAINS LOWEST ADDRESS IN THE ALLOCATED BLOCK
                        ;       FIRST TWO BYTES OF ALLOCATED BLOCK = BLOCK SIZE
                        ;
B33E A0 02      MALLOC  LDY     #S1H-DTAB       ; ACC = S1H ...
B340 20 16 B3           JSR     DLOADA

B343 A0 0A              LDY     #MEMB-DTAB      ; ... + MEMB.
B345 20 20 B3           JSR     DADDA

B348 A0 04              LDY     #S2L-DTAB       ; COMPARE ACC WITH S2L.
B34A 20 2A B3           JSR     DCMPA
```

```
B34D 90 03              BCC     *+5
B34F 4C D6 B3           JMP     MAL300

B352 A2 08              LDX     #MEMA-DTAB      ; SEE IF ALLOCATION IN REGION #1 OR #2.
B354 A0 04              LDY     #S2L-DTAB
B356 20 CE B2           JSR     DCMPI
B359 B0 35              BCS     MAL100          ; REGION #2.

                ; ALLOCATE FROM REGION #1

B35B A2 0C              LDX     #MSP-DTAB       ; MSP = MEMA.
B35D A0 08              LDY     #MEMA-DTAB
B35F 20 E9 B2           JSR     DMOVI

B362 A2 0E              LDX     #MDP-DTAB       ; MDP = MEMA ...
B364 20 E9 B2           JSR     DMOVI

B367 A0 0A              LDY     #MEMB-DTAB      ; ... + MEMB.
B369 20 F4 B2           JSR     DADDI

B36C A2 10              LDX     #MBC-DTAB       ; MBC = S1H ...
B36E A0 02              LDY     #S1H-DTAB
B370 20 E9 B2           JSR     DMOVI

B373 A0 08              LDY     #MEMA-DTAB      ; ... - MEMA.
B375 20 04 B3           JSR     DSUBI

B378 A2 02              LDX     #S1H-DTAB       ; S1H = ACC (= S1H + MEMB).
B37A 20 1B B3           JSR     DSTORA

B37D 20 84 B4           JSR     MOVDA           ; MOVE DATA UPWARD.
                ; FREE MEM CALC
B380 A5 84              LDA     S2L
B382 85 92              STA     ACC
B384 A5 85              LDA     S2L+1
B386 85 93              STA     ACC+1
B388 A0 02              LDY     #S1H-DTAB
B38A 20 25 B3           JSR     DSUBA

B38D 4C C8 B3           JMP     MAL200

                ; ALLOCATE IN REGION #2

B390 A2 0C      MAL100  LDX     #MSP-DTAB       ; MSP = S2L.
B392 A0 04              LDY     #S2L-DTAB
B394 20 E9 B2           JSR     DMOVI

B397 A2 10              LDX     #MBC-DTAB       ; MBC = MEMA ...
B399 A0 08              LDY     #MEMA-DTAB
B39B 20 E9 B2           JSR     DMOVI

B39E A0 04              LDY     #S2L-DTAB       ; ... - S2L.
B3A0 20 04 B3           JSR     DSUBI

B3A3 A2 04              LDX     #S2L-DTAB       ; S2L = S2L - MEMB.
B3A5 A0 0A              LDY     #MEMB-DTAB
B3A7 20 04 B3           JSR     DSUBI

B3AA A2 0E              LDX     #MDP-DTAB       ; MDP = S2L (NEW VALUE).
B3AC A0 04              LDY     #S2L-DTAB
```

```
B3AE 20 E9 B2              JSR     DMOVI

B3B1 A2 08                 LDX     #MEMA-DTAB      ; MEMA = MEMA - MEMB.
B3B3 A0 0A                 LDY     #MEMB-DTAB
B3B5 20 04 B3              JSR     DSUBI

B3B8 20 5F B4              JSR     MOVIA           ; MOVE DATA DOWNWARD.
                    ; FREE MEM CALC
B3BB A5 84                 LDA     S2L
B3BD 85 92                 STA     ACC
B3BF A5 85                 LDA     S2L+1
B3C1 85 93                 STA     ACC+1
B3C3 A0 02                 LDY     #S1H-DTAB
B3C5 20 25 B3              JSR     DSUBA

B3C8 A0 00         MAL200  LDY     #0              ; MOVE BLOCK SIZE TO BLOCK.
B3CA A5 8A                 LDA     MEMB
B3CC 91 88                 STA     (MEMA),Y
B3CE C8                    INY
B3CF A5 8B                 LDA     MEMB+1
B3D1 91 88                 STA     (MEMA),Y

B3D3 A9 00                 LDA     #0              ; SET CC FOR NORMAL EXIT.
B3D5 60                    RTS

B3D6 A9 FF         MAL300  LDA     #$FF            ; SET CC FOR ERROR EXIT.
B3D8 60                    RTS
```

```
                    ;
                    ; MDEALL -- MEMORY DEALLOCATE
                    ;
                    ; CALLING SEQUENCE:
                    ;
                    ;       'MEMA' = ADDRESS OF BLOCK TO DEALLOCATE
                    ;       FIRST 2 BYTES OF BLOCK = SIZE OF BLOCK
                    ;
                    ;       JSR     MDEALL
                    ;
                    ;       'MEMA' = ADDRESS OF BLOCK FOLLOWING DEALLOCATED BLOCK (AFTER DEALL)
                    ;
B3D9 A0 00          MDEALL  LDY     #0              ; GET SIZE OF BLOCK TO MEMB.
B3DB B1 88                  LDA     (MEMA),Y
B3DD 85 8A                  STA     MEMB
B3DF C8                     INY
B3E0 B1 88                  LDA     (MEMA),Y
B3E2 85 8B                  STA     MEMB+1

B3E4 A2 08                  LDX     #MEMA-DTAB      ; SEE IF IN REGION #1 OR #2.
B3E6 A0 04                  LDY     #S2L-DTAB
B3E8 20 CE B2               JSR     DCMPI
B3EB B0 39                  BCS     MDA100          ; REGION #2.

                    ; DEALLOCATE FROM REGION #1.

B3ED A2 0C                  LDX     #MSP-DTAB       ; MSP = MEMA ...
B3EF A0 08                  LDY     #MEMA-DTAB
B3F1 20 E9 B2               JSR     DMOVI
```

```
B3F4 A0 0A              LDY     #MEMB-DTAB      ; ... + MEMB.
B3F6 20 F4 B2           JSR     DADDI

B3F9 A2 10              LDX     #MBC-DTAB       ; MBC = S1H ...
B3FB A0 02              LDY     #S1H-DTAB
B3FD 20 E9 B2           JSR     DMOVI

B400 A0 0C              LDY     #MSP-DTAB       ; ... - MSP.
B402 20 04 B3           JSR     DSUBI

B405 A2 02              LDX     #S1H-DTAB       ; S1H = S1H - MEMB.
B407 A0 0A              LDY     #MEMB-DTAB
B409 20 04 B3           JSR     DSUBI

B40C A2 0E              LDX     #MDP-DTAB       ; MDP = MEMA.
B40E A0 08              LDY     #MEMA-DTAB
B410 20 E9 B2           JSR     DMOVI

B413 20 5F B4           JSR     MOVIA           ; MOVE DATA DOWNWARD.
                ; FREE MEM CALC
B416 A5 84              LDA     S2L
B418 85 92              STA     ACC
B41A A5 85              LDA     S2L+1
B41C 85 93              STA     ACC+1
B41E A0 02              LDY     #S1H-DTAB
B420 20 25 B3           JSR     DSUBA

B423 4C 5E B4           JMP     MDA200

                ; DEALLOCATE MEMORY IN REGION #2

B426 A2 0C      MDA100  LDX     #MSP-DTAB       ; MSP = S2L.
B428 A0 04              LDY     #S2L-DTAB
B42A 20 E9 B2           JSR     DMOVI

B42D A2 10              LDX     #MBC-DTAB       ; MBC = MEMA ...
B42F A0 08              LDY     #MEMA-DTAB
B431 20 E9 B2           JSR     DMOVI

B434 A0 04              LDY     #S2L-DTAB       ; ... - S2L.
B436 20 04 B3           JSR     DSUBI

B439 A2 04              LDX     #S2L-DTAB       ; S2L = S2L + MEMB.
B43B A0 0A              LDY     #MEMB-DTAB
B43D 20 F4 B2           JSR     DADDI

B440 A2 0E              LDX     #MDP-DTAB       ; MDP = S2L (NEW VALUE).
B442 A0 04              LDY     #S2L-DTAB
B444 20 E9 B2           JSR     DMOVI

B447 A2 08              LDX     #MEMA-DTAB      ; MEMA = MEMA + MEMB.
B449 A0 0A              LDY     #MEMB-DTAB
B44B 20 F4 B2           JSR     DADDI

B44E 20 84 B4           JSR     MOVDA           ; MOVE DATA UPWARD.

                ; FREE MEM CALC
B451 A5 84              LDA     S2L
B453 85 92              STA     ACC
B455 A5 85              LDA     S2L+1
```

```
B457 85 93                  STA     ACC+1
B459 A0 02                  LDY     #S1H-DTAB
B45B 20 25 B3               JSR     DSUBA
B45E 60           MDA200    RTS
```

```
                        ;
                        ; MOVE UTILITIES FOR MEMORY MANAGEMENT
                        ;
                        ; MOVE BLOCKS OF DATA WITH EITHER INCREASING OR DECREASING ADDRESS
                        ;
                        ; THREE VARIABLES CONTROL THE MOVE ROUTINES:
                        ;
                        ;       'MSP' CONTAINS POINTER TO SOURCE DATA LOCATION
                        ;       'MDP' CONTAINS POINTER TO DESTINATION DATA LOCATION
                        ;       'MBC' CONTAINS THE NUMBER OF BYTES TO MOVE
                        ;
                        ;
                        ;
                        ; MOVIA -- MOVE DATA BLOCK WITH INCREASING ADDRESS
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'MSP', 'MDP' & 'MBC' SETUP
                        ;
                        ;       JSR     MOVIA
                        ;
B45F A5 90              MOVIA   LDA     MBC             ; SEE IF BYTE COUNT = ZERO.
B461 AA                         TAX                     ; SAVE LSB OF BYTE COUNT.
B462 05 91                      ORA     MBC+1
B464 F0 1D                      BEQ     MVI090          ; ZERO -- NOTHING TO DO.

B466 A0 00                      LDY     #0              ; INDEX TO DATA BLOCK.

B468 B1 8C              MVI010  LDA     (MSP),Y         ; MOVE DATA.
B46A 91 8E                      STA     (MDP),Y
B46C C8                         INY                     ; BUMP INDEX.
B46D D0 04                      BNE     MVI020          ; NO PAGE WRAP.

B46F E6 8D                      INC     MSP+1           ; PAGE WRAP -- BUMP POINTER VARIABLES.
B471 E6 8F                      INC     MDP+1

B473 CA                 MVI020  DEX                     ; DONE?
B474 D0 04                      BNE     MVI030          ; NO.

B476 A5 91                      LDA     MBC+1           ; NOT SURE -- CHECK FURTHER.
B478 F0 09                      BEQ     MVI090          ; YES -- DONE.

B47A E0 FF              MVI030  CPX     #$FF            ; MAINTAIN D.P. BYTE COUNT.
B47C D0 EA                      BNE     MVI010

B47E C6 91                      DEC     MBC+1           ; BORROW FROM MSB.
B480 4C 68 B4                   JMP     MVI010

B483 60                 MVI090  RTS


                        ;
                        ; MOVDA -- MOVE DATA BLOCK WITH DECREASING ADDRESS
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       'MSP', 'MDP', & 'MBC' SETUP
                        ;       JSR     MOVDA
```

```
                    ;
B484 A5 90    MOVDA LDA     MBC             ; SETUP BYTE COUNT . . .
B486 AA             TAX
B487 A8             TAY                     ; . . . AND DATA INDEX.
B488 05 91          ORA     MBC+1           ; TEST FOR ZERO BYTE COUNT.
B48A F0 24          BEQ     MVD090          ; ZERO -- NOTHING TO DO.

B48C 18             CLC                     ; ADJUST POINTERS FOR START.
B48D A5 8D          LDA     MSP+1
B48F 65 91          ADC     MBC+1
B491 85 8D          STA     MSP+1

B493 18             CLC
B494 A5 8F          LDA     MDP+1
B496 65 91          ADC     MBC+1
B498 85 8F          STA     MDP+1

B49A 88       MVD010 DEY                    ; DECREMENT INDEX.
B49B C0 FF          CPY     #$FF            ; WRAP?
B49D D0 06          BNE     MVD020          ; NO.

B49F C6 91          DEC     MBC+1           ; YES -- DECREMENT ALL POINTERS (MSB).
B4A1 C6 8D          DEC     MSP+1
B4A3 C6 8F          DEC     MDP+1

B4A5 B1 8C    MVD020 LDA    (MSP),Y         ; MOVE A DATA BYTE.
B4A7 91 8E          STA     (MDP),Y

B4A9 CA             DEX                     ; DONE?
B4AA D0 EE          BNE     MVD010          ; NO -- CONTINUE.

B4AC A5 91          LDA     MBC+1           ; NOT SURE -- CHECK FURTHER.
B4AE D0 EA          BNE     MVD010          ; NO -- CONTINUE.

B4B0 60       MVD090 RTS                    ; YES -- RETURN.
                     *=V
086F          FRMLST *=*+959
0C2E          OBJLST *=*+1
0C2F          BACLST *=*+16
0C3F          BLSTLN *=*+1
0C40          OLSTLN *=*+1
0C41          MOVLEN *=*+1
                     .END
```

SYMBOL TABLE

ACC	0092	ACTION	B205	ADDCOR	030E	ADDFRM	A58F
ADDLP	A5C9	ADRESS	0064	AFP	D800	ALBAC	A4B6
ALLPOT	D208	ANTIC	D400	APPEND	0001	APPMHI	000E
ATACHR	02FB	ATAN	BE43	ATRACT	004D	AUDC1	D201
AUDC2	D203	AUDC3	D205	AUDC4	D207	AUDCTL	D208
AUDF1	D200	AUDF2	D202	AUDF3	D204	AUDF4	D206
BACACC	ADA2	BACCTR	080E	BACLST	0C2F	BACSEG	ABF5
BADIOC	0086	BADMOD	0091	BAKGND	0003	BB	085B
BELELN	0002	BFENHI	0035	BFENLO	0034	BITMSK	006E
BITOFF	0831	BKPLOP	AF8B	BLDSCN	AF77	BLIM	028A
BLKBDV	E471	BLNKPT	00A8	BLSTLN	0C3F	BOOT?	0009
BOOTAD	0242	BOTSCR	02BF	BOTTOM	0857	BPTR	003D
BRKABT	0080	BRKKEY	0011	BSLP	AC0D	BUFADR	0015
BUFCNT	006B	BUFRFL	0038	BUFRHI	0033	BUFRLO	0032
BUFSTR	006C	BUILD	A7E8	BYTEND	0851	BYTSTR	084D
BYTWID	0853	CASBUF	03FD	CASETV	E440	CASFLG	030F
CASINI	0002	CASORG	EF41	CASS	A2B4	CASSAV	A8AD
CASSBT	004B	CASSET	0043	CASSRD	A9AE	CAUX1	023C
CAUX2	023D	CBAUDH	02EF	CBAUDL	02EE	CCOMND	023B
CDEVIC	023A	CDTMA1	0226	CDTMA2	0228	CDTMF3	022A
CDTMF4	022C	CDTMF5	022E	CDTMV1	0218	CDTMV2	021A
CDTMV3	021C	CDTMV4	021E	CDTMV5	0220	CFRMLP	A073
CH	02FC	CH1	02F2	CHACT	02F3	CHACTL	D401
CHAR	02FA	CHBAS	02F4	CHBASE	D409	CHGFLG	0804
CHGFRM	A5DA	CHGLP	A5F3	CHKDWN	B264	CHKERR	008F
CHKLFT	B27D	CHKPF0	A530	CHKPF1	A539	CHKPF2	A542
CHKPF3	A54B	CHKRIT	B29C	CHKSNT	003B	CHKSUM	0031
CHRORG	E000	CHRPTR	00AA	CIOCHR	002F	CIOINV	E46E
CIOORG	E4A6	CIOV	E456	CIX	00F2	CKEY	004A
CLOOP2	B0E3	CLOSE	000C	CLRLOD	A11C	CLRLUP	B0C7
CLSLCT	0807	CNTMAK	A3DB	COLAC	0072	COLBK	D01A
COLCRS	0055	COLDST	0244	COLDSV	E477	COLINC	007A
COLOR0	02C4	COLOR1	02C5	COLOR2	02C6	COLOR3	02C7
COLOR4	02C8	COLOUR	0866	COLPF0	D016	COLPF1	D017
COLPF2	D018	COLPF3	D019	COLPM0	D012	COLPM1	D013
COLPM2	D014	COLPM3	D015	COLR	086A	COLRSH	004F
COLRTB	ACF4	CONSOL	D01F	COS	BD73	COUNTR	007E
CPATTB	B1CE	CPLRBD	AEA8	CR	009B	CRETRY	0036
CRITIC	0042	CRSINH	02F0	CRSROR	008D	CSCNBD	AF64
CSOPIV	E47D	CSTART	A001	CSTAT	0288	CSTATE	A186
CTIA	D000	CURFRM	00B6	CURPLR	0835	CURSOR	B1D2
CURSTT	0801	DADDA	B320	DADDI	B2F4	DATLEN	0003
DAUX1	030A	DAUX2	030B	DBSECT	0241	DBUFHI	0305
DBUFLO	0304	DBYTHI	0309	DBYTLO	0308	DCB	0300
DCM090	B2DA	DCMPA	B32A	DCMPI	B2CE	DCOMND	0302
DDEVIC	0300	DEGFLG	00FB	DEGON	0006	DELBAK	AB2B
DELETE	0021	DELTAC	0077	DELTAR	0076	DERROR	0090
DFLAGS	0240	DIGRT	00F1	DINDEX	0057	DIRECT	0002
DISK	0044	DISKIV	E450	DISPLY	0053	DLISTH	D403
DLISTL	D402	DLOADA	B316	DLPTR	00AE	DMACTL	D400
DMASK	02A0	DMOVI	B2E9	DMPY16	AEED	DMPY2	AEF9
DMPY4	AEF5	DMPY8	AEF1	DNACK	008B	DOSINI	000C
DOSVEC	000A	DRAWLN	0011	DRETRY	0037	DRKMSK	004E
DS1H	009E	DS1L	009C	DS1LP	AF18	DS1TS1	AF16
DS2H	00A2	DS2L	00A0	DSC030	B2DB	DSC032	B2DF
DSC035	B2E3	DSC037	B2E5	DSKFMS	0018	DSKINV	E453
DSKORG	EDEA	DSKTIM	0246	DSKUTL	001A	DSPFLG	02FE
DSTAT	004C	DSTATS	0303	DSTORA	B31B	DSUBA	B325
DSUBI	B304	DTAB	0080	DTIMLO	0306	DTSTRT	0004

MICRO FLICK - J. HUETHER

DUNIT	0301	DUNUSE	0307	DVSTAT	02EA	ECHO	A323
EDITRV	E400	EEXP	00ED	ELOOP	AC92	ENDBSQ	AC61
ENDFSQ	A660	ENDMOD	A892	ENDPLP	AF86	ENDPLR	A482
ENDPP	AFE1	ENDPT	0074	ENDSPD	ACD9	ENDX	085C
ENDY	085D	EOFERR	0088	ERRCHK	A1A3	ERRFLG	023F
ERSCRN	AC76	ERSMSL	B2C5	ESCFLG	02A2	ESIGN	00EF
EVBHND	ACFE	EXITP	A412	EXP	DDC0	EXP10	DDCC
EXTADD	A5D7	EXTCHG	A606	EXTERS	ACA9	FADD	DA66
FASC	D8E6	FBYTWD	084C	FCHRFL	00F0	FCNKEY	A34B
FDIV	DB28	FELELN	0010	FEOF	003F	FIL	AD7B
FIL010	AD84	FIL020	AD8E	FIL030	AD95	FIL090	AD9E
FILDAT	02FD	FILFLG	02B7	FILLIN	0012	FILSCN	B0C5
FILVAL	0867	FLDOP	DD8D	FLDOR	DD89	FLD1P	DD9C
FLD1R	DD98	FLDLEN	0008	FLPTR	00FC	FMOVE	DDB6
FMSZPG	0043	FMUL	DADB	FNCNOT	0092	FORMAT	0022
FPI	D9D2	FPREC	0006	FPSCR	05E6	FPSCR1	05EC
FPTR2	00FE	FR0	00D4	FR1	00E0	FR2	00E6
FRE	00DA	FREQ	0040	FRMACC	AF49	FRMCTR	080C
FRMERR	008C	FRMLST	086F	FRMSEQ	A609	FRX	00EC
FSCR	05E6	FSCR1	05EC	FSTOP	DDAB	FSTOR	DDA7
FSUB	DA60	FTYPE	003E	GETCHR	0007	GETKEE	B071
GETREC	0005	GLBABS	02E0	GPRIOR	026F	GRACTL	D01D
GRAFM	D011	GRAFP0	D00D	GRAFP1	D00E	GRAFP2	D00F
GRAFP3	D010	GXMISL	0580	GXP0	0600	GXP1	0680
GXP2	0700	GXP3	0780	GXPM	0400	HATABS	031A
HDELTA	083B	HEIGHT	0829	HITCLR	D01E	HOLD1	0051
HOLD2	029F	HOLD3	029D	HOLD4	02BC	HOLD5	02BD
HOLDCH	007C	HORMO	0815	HORPO	0811	HPOSM0	D004
HPOSM1	D005	HPOSM2	D006	HPOSM3	D007	HPOSP0	D000
HPOSP1	D001	HPOSP2	D002	HPOSP3	D003	HSCROL	D404
HWID	0808	ICAX1	034A	ICAX1Z	002A	ICAX2	034B
ICAX2Z	002B	ICBAH	0345	ICBAHZ	0025	ICBAL	0344
ICBALZ	0024	ICBLH	0349	ICBLHZ	0029	ICBLL	0348
ICBLLZ	0028	ICCOM	0342	ICCOMT	0017	ICCOMZ	0022
ICDNO	0341	ICDNOZ	0021	ICHID	0340	ICHIDZ	0020
ICIDNO	002E	ICPTH	0347	ICPTHZ	0027	ICPTL	0346
ICPTLZ	0026	ICSPR	034C	ICSPRZ	002C	ICSTA	0343
ICSTAZ	0023	IFP	D9AA	ILSKYB	A55D	ILSKYL	A6DF
ILSKYP	A402	ILSKYS	A658	INBUFF	00F3	INIT	A000
INKCLR	083A	INKPF	A51A	INKPOS	AE76	INSCLR	0020
INSDAT	007D	INTABS	0200	INTEMP	022D	INTINV	E46B
INTORG	E6D5	INTZBS	0010	INVFLG	02B6	IOCB	0340
IOCBAS	0020	IOCBSZ	0010	IOCFRE	00FF	IRGEN	D20E
IRGST	D20E	ISKEYP	A40A	JDON	B2B5	JMPTBL	A1B1
JOYRD	B21E	KBCODE	D209	KBD	004B	KBDORG	F3E4
KEEPON	A8AA	KEYBDV	E420	KEYBRD	A2AE	KEYDEL	02F1
LB	0858	LBFEND	05FF	LBPR1	057E	LBPR2	057F
LBUFF	0580	LEDGE	0002	LEFT	0854	LEXIT2	A6E7
LINBUF	0247	LINZBS	0000	LMARGN	0052	LOC1	A685
LOC123	A79E	LOC222	A744	LOC333	A747	LOCKFL	0023
LOCLP	A6AE	LOCPLR	A66C	LOG	DECD	LOG10	DED1
LOGCOL	0063	LOGMAP	02B2	LOOP79	AC85	LOOP80	AC7B
LOOP88	AC65	LOOP89	AB1F	LOOP90	AB13	LOOP92	A7BE
LOOP93	A627	LOOP94	A5EB	LOOP95	A5BB	LOOP96	AD62
LOOP98	A477	LOOP99	A152	LPENH	0234	LPENV	0235
LPTRTB	B0EC	LS1H	0096	LS1L	0094	LS1LP	AF00
LS1TS1	AEFE	LS2H	009A	LS2L	0098	LVMBAK	A565
M0PF	D000	M0PL	D008	M123	A3C3	M1PF	D001
M1PL	D009	M2PF	D002	M2PL	D00A	M3PF	D003
M3PL	D00B	MAIN	A17B	MAKBAK	A485	MAKBLP	A4FF

MAKPLP	A392	MAKPLR	A38A	MAL100	B390	MAL200	B3C8
MAL300	B3D6	MALLOC	B33E	MAXBAC	0008	MAXDEV	0021
MAXIOC	0080	MAXMSG	0008	MBC	0090	MCHK1	A853
MCHKM	A85F	MCHKRT	A87F	MDA100	B426	MDA200	B45E
MDEALL	B3D9	MDP	008E	MEMA	0088	MEMB	008A
MEMLO	02E7	MEMTOP	02E5	MESAGE	AECC	MINIT	B32F
MLTTMP	0066	MODE	0839	MODEM	004D	MODEND	0836
MODFLG	0864	MODLP	A837	MODPLR	A803	MONORG	F0E3
MOVDA	B484	MOVEIT	A6F7	MOVEUP	A779	MOVIA	B45F
MOVLEN	0C41	MPFCRS	A50D	MSG1	A1F5	MSG2	A214
MSG3	A226	MSG4	A238	MSG5	A24C	MSG6	A25E
MSG7	A27C	MSG8	A296	MSGTBL	A1E5	MSP	008C
MSPEED	0805	MSTPBD	A80D	MVD010	B49A	MVD020	B4A5
MVD090	B4B0	MVI010	B468	MVI020	B473	MVI030	B47A
MVI090	B483	MXDMOD	0010	NEWCOL	0061	NEWROW	0060
NLPLR	A674	NMIEN	D40E	NMIRES	D40F	NMIST	D40F
NOACTN	B204	NOALOW	A37F	NOALWE	ACA1	NOALWW	ACD1
NOCKSM	003C	NOEDGE	B1F2	NOFKEY	A387	NOHGHT	AE6B
NOKEY	A3A7	NOKEY2	A895	NONDEV	0082	NOTE	0026
NOTOPN	0085	NSIGN	00EE	NVALID	0084	NXTSTT	0803
OBJACC	AF2E	OBJCTR	080D	OBJLST	0C2E	OELELN	0002
OLDADR	005E	OLDCHR	005D	OLDCOL	005B	OLDHGT	082D
OLDROW	005A	OLDVMO	081D	OLDVPO	0819	OLSTLN	0C40
ONETBL	ACE4	OPEN	0003	OPNIN	0004	OPNINO	000C
OPNOT	0008	OPNTMP	0066	OUTRNG	AE9C	OVRRUN	008E
P	B1D2	P0	0004	P0PF	D004	P0PL	D00C
P1	0005	P111	A3F0	P1PF	D005	P1PL	D00D
P2	0006	P222	A3F6	P2PF	D006	P2PL	D00E
P3	0007	P3PF	D007	P3PL	D00F	PACTL	D302
PADDL0	0270	PADDL1	0271	PADDL2	0272	PADDL3	0273
PADDL4	0274	PADDL5	0275	PADDL6	0276	PADDL7	0277
PAKCOL	086B	PBCTL	D303	PBOT	080B	PBPNT	001D
PBTEND	0050	PBUFSZ	001E	PCHK0	A3E7	PCOLR0	02C0
PCOLR1	02C1	PCOLR2	02C2	PCOLR3	02C3	PENH	D40C
PENV	D40D	PIA	D300	PIXLEN	084F	PIXWID	084E
PLBND	0837	PLPLYR	AFCC	PLYARG	05E0	PLYEVL	DD40
PLYRTB	ACEC	PMBASE	D407	POFFLP	ABF7	POINT	0025
POKEY	D200	POKMSK	0010	PORTA	D300	PORTB	D301
POT0	D200	POT1	D201	POT2	D202	POT3	D203
POT4	D204	POT5	D205	POT6	D206	POT7	D207
POTGO	D20B	PPLOP	AFB7	PPLPLP	AF7F	PRBND	0838
PRINTR	0050	PRINTV	E430	PRIOR	D01B	PRNBUF	03C0
PRNORG	EE78	PRTEND	00A0	PRVFRM	00C6	PRVOPN	0081
PRVPTA	086D	PRVSTT	0802	PRVTRO	0865	PTEMP	001F
PTIMOT	001C	PTINDX	0862	PTOP	080A	PTR	00B4
PTRIG0	027C	PTRIG1	027D	PTRIG2	027E	PTRIG3	027F
PTRIG4	0280	PTRIG5	0281	PTRIG6	0282	PTRIG7	0283
PUPDAT	00A4	PUTICH	B05F	PUTCHR	000B	PUTREC	0009
PWRTPX	B084	PXOOTB	B1CA	PX11TB	B1C6	PZERPX	B0A6
PZJUMP	00B0	QRJUS	B183	RADFLG	00FB	RADON	0000
RAMLO	0004	RAMPTR	00AC	RAMSIZ	02E4	RAMTOP	006A
RANDOM	D20A	RB	085A	RBLOKV	E47A	RDKEY	A333
RDONLY	0087	READPX	B14C	RECVDN	0039	REDGE	0027
RENAME	0020	RIGHT	0856	RITEND	0050	RJUSLP	B17A
RLP1	A7F8	RMARGN	0053	ROWAC	0070	ROWCRS	0054
ROWINC	0079	RPTCNT	086E	RTCL	0800	RTCLOK	0012
RUN	A7AD	RUNLP	A7B5	RUNSNC	A7C8	S1H	0082
S1L	0080	S1TDLP	AF24	S1TDS1	AF22	S1TLLP	AF0C
S1TLS1	AF0A	S2H	0086	S2L	0084	SAMEFM	A632
SAVADR	0068	SAVALL	AD01	SAVCRD	ADC9	SAVIO	0316

MICRO FLICK - J. HUETHER

SAVKEY	0806	SAVMSC	0058	SAVPLR	ADD9	SCNEND	0028
SCNPTR	00A6	SCREDT	0045	SCRENV	E410	SCRFLG	02BB
SCRMEM	0093	SCRWID	0014	SDLSTH	0231	SDLSTL	0230
SDMCTL	022F	SDSPLY	A2B1	SENDEV	E468	SERIN	D20D
SEROUT	D20D	SETFF	ABDF	SETSPD	ACAC	SETVBV	E45C
SHFAMT	006F	SHFLOK	02BE	SHOWB	AC32	SHOWFM	A62F
SIN	BD81	SIOINV	E465	SIOORG	E944	SIOV	E459
SIZEM	D00C	SIZEP0	D008	SIZEP1	D009	SIZEP2	D00A
SIZEP3	D00B	SIZLP	A129	SKCTL	D20F	SKEYP	A3E4
SKRES	D20A	SKSTAT	D20F	SOFFND	0850	SOFFST	0852
SOUNDR	0041	SPDLP	A142	SPECIL	000E	SPFCLR	A551
SQR	BEB1	SRTIMR	022B	SSFLAG	02FF	SSKCTL	0232
STACKP	0318	STATBL	A2B7	STATIS	000D	STATUS	0030
STICK0	0278	STICK1	0279	STICK2	027A	STICK3	027B
STIMER	D209	STRIG0	0284	STRIG1	0285	STRIG2	0286
STRIG3	0287	SUBKYB	A52D	SUBKYL	A6BD	SUBKYP	A3C6
SUBKYS	A641	SUBKYW	ACBD	SUBTMP	029E	SUCCES	0001
SVCRFM	A611	SWPFLG	007B	SYSVBV	E45F	TABMAP	02A3
TAVAIL	A3CF	TB	0859	TEMP	023E	TEMP1	0312
TEMP2	0314	TEMP3	0315	TIMER1	030C	TIMER2	0310
TIMFLG	0317	TIMOUT	008A	TINDEX	0293	TMP1	0868
TMP2	0869	TMPCHR	0050	TMPCOL	02B9	TMPLBT	02A1
TMPROW	02BB	TMPX1	029C	TOP	0855	TRAMSZ	0006
TRIG0	D010	TRIG1	D011	TRIG2	D012	TRIG3	D013
TRNRCD	0089	TRYNXT	A874	TSTAT	0319	TSTDAT	0007
TVSEND	AE43	TXTCOL	0291	TXTMSC	0294	TXTOLD	0296
TXTROW	0290	UNLOCK	0024	UPDFLG	0863	UPDTBP	AB94
UPDTRF	ABC2	USAREA	0480	V	086F	VBFLAG	084B
VBHND	ACF8	VBREAK	0206	VCOUNT	D40B	VCTABL	E480
VDELAY	D01C	VDELTA	0843	VDSLST	0200	VECTBL	E400
VIMIRQ	0216	VINTER	0204	VKEYBD	0208	VPOSMO	0825
VPOSP0	0821	VPRCED	0202	VRTSIZ	AE14	VSCROL	D405
VSERIN	020A	VSEROC	020E	VSEROR	020C	VSIZLP	AE2F
VTIMR1	0210	VTIMR2	0212	VTIMR4	0214	VVBLKD	0224
VVBLKI	0222	VWID	0809	WARMST	0008	WARMSV	E474
WFVBFG	AE9D	WMODE	0289	WRITE	AFE2	WRITPX	B187
WRONLY	0083	WRTEMP	086C	WRTMSL	B2B6	WRTONE	AE99
WSYNC	D40A	X0	0008	X1	0009	X2	000A
X3	000B	XBOX	085E	XFER11	A616	XITVBV	E462
XMTDON	003A	XPOS	080F	XXXX	ABC7	Y0	000C
Y1	000D	Y2	000E	Y3	000F	YBOX	0860
YPOS	0810	YYYY	ABE3	Z	00D6	ZERMEM	AA97
ZERPLR	AD6D	ZERTBL	ACDC	ZIOCB	0020	ZTEMP1	00F5
ZTEMP3	00F9	ZTEMP4	00F7				