ATARI® 400/800™

ATARI® HOME COMPUTER SYSTEM

OPERATING SYSTEM SOURCE LISTING

ATARI®
A Warner Communications Company

COPYRIGHT 1982, ATARI, INC.
ALL RIGHTS RESERVED

TO ALL PERSONS RECEIVING THIS DOCUMENT

Reproduction is forbidden without the specific written permission of ATARI, INC. Sunnyvale, CA 94086. No right to reproduce this document, nor the subject matter thereof, is granted unless by written agreement with, or written permission from the Corporation.

Every effort has been made to ensure that this manual accurately documents this product of the ATARI Home Computer Division. However, due to the ongoing improvement and update of the computer software and hardware, ATARI, INC. cannot guarantee the accuracy of printed material after the date of publication and disclaims liability for changes, errors, or omissions.

```
                        LIST X
                        ;THIS IS THE MODIFIED SEPTEMBER ATARI 400/800 COMPUTER OPERATING
                        ; SYSTEM LISTING, MODIFIED TO ASSEMBLE ON THE MICROTEC CROSS
                        ; ASSEMBLER.
                        ; THIS VERSION IS THE ONE WHICH WAS BURNED INTO ROM.
                        ; THERE IS A RESIDUAL PIECE OF CODE WHICH IS FOR LNBUG.  THIS
                        ; IS AT LOCATION $9000 WHICH IS NOT IN ROM.
                        ;
                        ; THIS IS THE REVISION B EPROM VERSION
```

```
                                .PAGE
                        ;
                        ;
                        ;       COLLEEN OPERATING SYSTEM EQUATE FILE
                        ;
                        ;       NTSC/PAL ASSEMBLY FLAG
                        ;
0000                    PALFLG  =       0               ;0 = NTSC   1 = PAL
                        ;
                        ;
                        ;       MODULE ORIGIN TABLE
                        ;
E000                    CHRORG  =       $E000           ;CHARACTER SET
E400                    VECTBL  =       $E400           ;VECTOR TABLE
E480                    VCTABL  =       $E480           ;RAM VECTOR INITIAL VALUE TABLE
E4A6                    CIOORG  =       $E4A6           ;CENTRAL I/O HANDLER
E6D5                    INTORG  =       $E6D5           ;INTERRUPT HANDLER
E944                    SIOORG  =       $E944           ;SERIAL I/O DRIVER
EDEA                    DSKORG  =       $EDEA           ;DISK HANDLER
EE78                    PRNORG  =       $EE78           ;PRINTER HANDLER
EF41                    CASORG  =       $EF41           ;CASSETTE HANDLER
F0E3                    MONORG  =       $F0E3           ;MONITOR/POWER UP MODULE
F3E4                    KBDORG  =       $F3E4           ;KEYBOARD/DISPLAY HANDLER
                        ;
                        ;
                        ;
                        ;
                        ;       VECTOR TABLE
                        ;
                        ;HANDLER ENTRY POINTS ARE CALLED OUT IN THE FOLLOWING VECTOR
                        ;TABLE.  THESE ARE THE ADDRESSES MINUS ONE.
                        ;
                        ;
                        ;EXAMPLE FOR EDITOR
                        ;
                        ;     E400      OPEN
                        ;        2      CLOSE
                        ;        4      GET
                        ;        6      PUT
                        ;        8      STATUS
                        ;        A      SPECIAL
                        ;        C      JUMP TO POWER ON INITIALIZATION ROUTINE
                        ;        F      NOT USED
                        ;
                        ;
E400                    EDITRV  =       $E400           ;EDITOR
E410                    SCRENV  =       $E410           ;TELEVISION SCREEN
E420                    KEYBDV  =       $E420           ;KEYBOARD
E430                    PRINTV  =       $E430           ;PRINTER
E440                    CASETV  =       $E440           ;CASSETTE
                        ;
                        ;       JUMP VECTOR TABLE
                        ;
                        ;THE FOLLOWING IS A TABLE OF JUMP INSTRUCTIONS
```

```
                                ;TO VARIOUS ENTRY POINTS IN THE OPERATING SYSTEM.
                                ;
E450                            DISKIV  =       $E450           ;DISK INITIALIZATION
E453                            DSKINV  =       $E453           ;DISK INTERFACE
E456                            CIOV    =       $E456           ;CENTRAL INPUT OUTPUT ROUTINE
E459                            SIOV    =       $E459           ;SERIAL INPUT OUTPUT ROUTINE
E45C                            SETVBV  =       $E45C           ;SET SYSTEM TIMERS ROUTINE
E45F                            SYSVBV  =       $E45F           ;SYSTEM VERTICAL BLANK CALCULATIONS
E462                            XITVBV  =       $E462           ;EXIT VERTICAL BLANK CALCULATIONS
E465                            SIOINV  =       $E465           ;SERIAL INPUT OUTPUT INITIALIZATION
E468                            SENDEV  =       $E468           ;SEND ENABLE ROUTINE
E46B                            INTINV  =       $E46B           ;INTERRUPT HANDLER INITIALIZATION
E46E                            CIOINV  =       $E46E           ;CENTRAL INPUT OUTPUT INITIALIZATION
E471                            BLKBDV  =       $E471           ;BLACKBOARD MODE
E474                            WARMSV  =       $E474           ;WARM START ENTRY POINT
E477                            COLDSV  =       $E477           ;COLD START ENTRY POINT
E47A                            RBLOKV  =       $E47A           ;CASSETTE READ BLOCK ENTRY POINT VECTOR
E47D                            CSOPIV  =       $E47D           ;CASSETTE OPEN FOR INPUT VECTOR
                                ;VCTABL = $E480
                                ;
                                ;
                                ; OPERATING SYSTEM EQUATES
                                ;
                                ; COMMAND CODES FOR IOCB
0003                            OPEN    =       3               ;OPEN FOR INPUT/OUTPUT
0005                            GETREC  =       5               ;GET RECORD (TEXT)
0007                            GETCHR  =       7               ;GET CHARACTER(S)
0009                            PUTREC  =       9               ;PUT RECORD (TEXT)
000B                            PUTCHR  =       $B              ;PUT CHARACTER(S)
000C                            CLOSE   =       $C              ;CLOSE DEVICE
000D                            STATIS  =       $D              ;STATUS REQUEST
000E                            SPECIL  =       $E              ;BEGINNING OF SPECIAL ENTRY COMMANDS
                                ;
                                ; SPECIAL ENTRY COMMANDS
0011                            DRAWLN  =       $11             ;DRAW LINE
0012                            FILLIN  =       $12             ;DRAW LINE WITH RIGHT FILL
0020                            RENAME  =       $20             ;RENAME DISK FILE
0021                            DELETE  =       $21             ;DELETE DISK FILE
0022                            FORMAT  =       $22             ;FORMAT
0023                            LOCKFL  =       $23             ;LOCK FILE TO READ ONLY
0024                            UNLOCK  =       $24             ;UNLOCK LOCKED FILE
0025                            POINT   =       $25             ;POINT SECTOR
0026                            NOTE    =       $26             ;NOTE SECTOR
00FF                            IOCFRE  =       $FF             ;IOCB "FREE"
                                ;
                                ; AUX1 EQUATES
                                ; () INDICATES WHICH DEVICES USE BIT
0001                            APPEND  =       $1              ;OPEN FOR WRITE APPEND (D), OR SCREEN READ (
0002                            DIRECT  =       $2              ;OPEN FOR DIRECTORY ACCESS (D)
0004                            OPNIN   =       $4              ;OPEN FOR INPUT (ALL DEVICES)
0008                            OPNOT   =       $8              ;OPEN FOR OUTPUT (ALL DEVICES)
000C                            OPNINO  =       OPNIN+OPNOT     ;OPEN FOR INPUT AND OUTPUT (ALL DEVICES)
0010                            MXDMOD  =       $10             ;OPEN FOR MIXED MODE (E,S)
0020                            INSCLR  =       $20             ;OPEN WITHOUT CLEARING SCREEN (E,S)
```

```
118                     ;
119                     ; DEVICE NAMES
120  0045               SCREDT  =       'E          ;SCREEN EDITOR (R/W)
121  004B               KBD     =       'K          ;KEYBOARD (R ONLY)
122  0053               DISPLY  =       'S          ;SCREEN DISPLAY (R/W)
123  0050               PRINTR  =       'P          ;PRINTER (W ONLY)
124  0043               CASSET  =       'C          ;CASSETTE
125  004D               MODEM   =       'M          ;MODEM
126  0044               DISK    =       'D          ;DISK (R/W)
127                     ;
128                     ; SYSTEM EOL (CARRIAGE RETURN)
129  009B               CR      =       $9B
130                     ;
131                     ;
132                     ;       OPERATING SYSTEM STATUS CODES
133                     ;
134  0001               SUCCES  =       $01         ;SUCCESSFUL OPERATION
135                     ;
136  0080               BRKABT  =       $80         ;BREAK KEY ABORT
137  0081               PRVOPN  =       $81         ;IOCB ALREADY OPEN
138  0082               NONDEV  =       $82         ;NON-EXISTANT DEVICE
139  0083               WRONLY  =       $83         ;IOCB OPENED FOR WRITE ONLY
140  0084               NVALID  =       $84         ;INVALID COMMAND
141  0085               NOTOPN  =       $85         ;DEVICE OR FILE NOT OPEN
142  0086               BADIOC  =       $86         ;INVALID IOCB NUMBER
143  0087               RDONLY  =       $87         ;IOCB OPENED FOR READ ONLY
144  0088               EOFERR  =       $88         ;END OF FILE
145  0089               TRNRCD  =       $89         ;TRUNCATED RECORD
146  008A               TIMOUT  =       $8A         ;PERIPHERAL DEVICE TIME OUT
147  008B               DNACK   =       $8B         ;DEVICE DOES NOT ACKNOWLEDGE COMMAND
148  008C               FRMERR  =       $8C         ;SERIAL BUS FRAMING ERROR
149  008D               CRSROR  =       $8D         ;CURSOR OVERRANGE
150  008E               OVRRUN  =       $8E         ;SERIAL BUS DATA OVERRUN
151  008F               CHKERR  =       $8F         ;SERIAL BUS CHECKSUM ERROR
152                     ;
153  0090               DERROR  =       $90         ;PERIPHERAL DEVICE ERROR (OPERATION NOT COMP
154  0091               BADMOD  =       $91         ;BAD SCREEN MODE NUMBER
155  0092               FNCNOT  =       $92         ;FUNCTION NOT IMPLEMENTED IN HANDLER
156  0093               SCRMEM  =       $93         ;INSUFICIENT MEMORY FOR SCREEN MODE
157                     ;
158                     ;
159                     ;
160                     ;
161                     ;
162                     ;
163                     ;       PAGE ZERO RAM ASSIGNMENTS
164                     ;
165                             *=$0000
166  0000               LINZBS: .RES    2           ;LINBUG RAM (WILL BE REPLACED BY MONITOR RAM
167                     ;
168                     ; THESE LOCATIONS ARE NOT CLEARED
169  0002               CASINI: .RES    2           ;CASSETTE INIT LOCATION
170  0004               RAMLO:  .RES    2           ;RAM POINTER FOR MEMORY TEST
171  0006               TRAMSZ: .RES    1           ;TEMPORARY REGISTER FOR RAM SIZE
```

```
     172  0007                    TSTDAT: .RES    1           ;RAM TEST DATA REGISTER
     173                          ;
     174                          ; CLEARED ON COLDSTART ONLY
     175  0008                    WARMST: .RES    1           ;WARM START FLAG
     176  0009                    BOOT?:  .RES    1           ;SUCCESSFUL BOOT FLAG
     177  000A                    DOSVEC: .RES    2           ;DISK SOFTWARE START VECTOR
     178  000C                    DOSINI: .RES    2           ;DISK SOFTWARE INIT ADDRESS
     179  000E                    APPMHI: .RES    2           ;APPLICATIONS MEMORY HI LIMIT
     180                          ;
     181                          ; CLEARED ON COLD OR WARM START
     182  0010                    INTZBS  =*                  ;INTERRUPT HANDLER
     183  0010                    POKMSK: .RES    1           ;SYSTEM MASK FOR POKEY IRQ ENABLE
     184  0011                    BRKKEY: .RES    1           ;BREAK KEY FLAG
     185  0012                    RTCLOK: .RES    3           ;REAL TIME CLOCK (IN 16 MSEC UNITS)
     186                          ;
     187  0015                    BUFADR: .RES    2           ;INDIRECT BUFFER ADDRESS REGISTER
     188                          ;
     189  0017                    ICCOMT: .RES    1           ;COMMAND FOR VECTOR
     190                          ;
     191  0018                    DSKFMS: .RES    2           ;DISK FILE MANAGER POINTER
     192  001A                    DSKUTL: .RES    2           ;DISK UTILITIES POINTER
     193                          ;
     194  001C                    PTIMOT: .RES    1           ;PRINTER TIME OUT REGISTER
     195  001D                    PBPNT:  .RES    1           ;PRINT BUFFER POINTER
     196  001E                    PBUFSZ: .RES    1           ;PRINT BUFFER SIZE
     197  001F                    PTEMP:  .RES    1           ;TEMPORARY REGISTER
     198                          ;
     199  0020                    ZIOCB   =*                  ;ZERO PAGE I/O CONTROL BLOCK
     200  0010                    IOCBSZ  =       16          ;NUMBER OF BYTES PER IOCB
     201  0080                    MAXIOC  =       8*IOCBSZ    ;LENGTH OF THE IOCB AREA
     202  0020                    IOCBAS  =*
     203  0020                    ICHIDZ: .RES    1           ;HANDLER INDEX NUMBER (FF = IOCB FREE)
     204  0021                    ICDNOZ: .RES    1           ;DEVICE NUMBER (DRIVE NUMBER)
     205  0022                    ICCOMZ: .RES    1           ;COMMAND CODE
     206  0023                    ICSTAZ: .RES    1           ;STATUS OF LAST IOCB ACTION
     207  0024                    ICBALZ: .RES    1           ;BUFFER ADDRESS LOW BYTE
     208  0025                    ICBAHZ: .RES    1
     209  0026                    ICPTLZ: .RES    1           ;PUT BYTE ROUTINE ADDRESS - 1
     210  0027                    ICPTHZ: .RES    1
     211  0028                    ICBLLZ: .RES    1           ;BUFFER LENGTH LOW BYTE
     212  0029                    ICBLHZ: .RES    1
     213  002A                    ICAX1Z: .RES    1           ;AUXILIARY INFORMATION FIRST BYTE
     214  002B                    ICAX2Z: .RES    1
     215  002C                    ICSPRZ: .RES    4           ;TWO SPARE BYTES (CIO LOCAL USE)
     216  002E                    ICIDNO  =       ICSPRZ+2    ;IOCB NUMBER X 16
     217  002F                    CIOCHR  =       ICSPRZ+3    ;CHARACTER BYTE FOR CURRENT OPERATION
     218                          ;
     219  0030                    STATUS: .RES    1           ;INTERNAL STATUS STORAGE
     220  0031                    CHKSUM: .RES    1           ;CHECKSUM (SINGLE BYTE SUM WITH CARRY)
     221  0032                    BUFRLO: .RES    1           ;POINTER TO DATA BUFFER (LO BYTE)
     222  0033                    BUFRHI: .RES    1           ;POINTER TO DATA BUFFER (HI BYTE)
     223  0034                    BFENLO: .RES    1           ;NEXT BYTE PAST END OF THE DATA BUFFER (LO B
     224  0035                    BFENHI: .RES    1           ;NEXT BYTE PAST END OF THE DATA BUFFER (HI B
     225  0036                    CRETRY: .RES    1           ;NUMBER OF COMMAND FRAME RETRIES
```

```
0037                DRETRY: .RES   1          ;NUMBER OF DEVICE RETRIES
0038                BUFRFL: .RES   1          ;DATA BUFFER FULL FLAG
0039                RECVDN: .RES   1          ;RECEIVE DONE FLAG
003A                XMTDON: .RES   1          ;TRANSMISSION DONE FLAG
003B                CHKSNT: .RES   1          ;CHECKSUM SENT FLAG
003C                NOCKSM: .RES   1          ;NO CHECKSUM FOLLOWS DATA FLAG
                    ;
                    ;
003D                BPTR:   .RES   1
003E                FTYPE:  .RES   1
003F                FEOF:   .RES   1
0040                FREQ:   .RES   1
0041                SOUNDR: .RES   1          ;NOISY I/O FLAG. (ZERO IS QUIET)
0042                CRITIC: .RES   1          ;DEFINES CRITICAL SECTION (CRITICAL IF NON-Z
                    ;
0043                FMSZPG: .RES   7          ;DISK FILE MANAGER SYSTEM ZERO PAGE
                    ;
                    ;
004A                CKEY:   .RES   1          ;FLAG SET WHEN GAME START PRESSED
004B                CASSBT: .RES   1          ;CASSETTE BOOT FLAG
004C                DSTAT:  .RES   1          ;DISPLAY STATUS
                    ;
004D                ATRACT: .RES   1          ;ATRACT FLAG
004E                DRKMSK: .RES   1          ;DARK ATRACT MASK
004F                COLRSH: .RES   1          ;ATRACT COLOR SHIFTER (EOR'ED WITH PLAYFIELD
                    ;
0002                LEDGE   =      2          ;LMARGN'S VALUE AT COLD START
0027                REDGE   =      39         ;RMARGN'S VALUE AT COLD START
0050                TMPCHR: .RES   1
0051                HOLD1:  .RES   1
0052                LMARGN: .RES   1          ;LEFT MARGIN (SET TO 1 AT POWER ON)
0053                RMARGN: .RES   1          ;RIGHT MARGIN (SET TO 38 AT POWER ON)
0054                ROWCRS: .RES   1          ;CURSOR COUNTERS
0055                COLCRS: .RES   2
0057                DINDEX: .RES   1
0058                SAVMSC: .RES   2
005A                OLDROW: .RES   1
005B                OLDCOL: .RES   2
005D                OLDCHR: .RES   1          ;DATA UNDER CURSOR
005E                OLDADR: .RES   2
0060                NEWROW: .RES   1          ;POINT DRAW GOES TO
0061                NEWCOL: .RES   2
0063                LOGCOL: .RES   1          ;POINTS AT COLUMN IN LOGICAL LINE
0064                ADRESS: .RES   2
0066                MLTTMP: .RES   2
0066                OPNTMP  =      MLTTMP     ;FIRST BYTE IS USED IN OPEN AS TEMP
0068                SAVADR: .RES   2
006A                RAMTOP: .RES   1          ;RAM SIZE DEFINED BY POWER ON LOGIC
006B                BUFCNT: .RES   1          ;BUFFER COUNT
006C                BUFSTR: .RES   2          ;EDITOR GETCH POINTER
006E                BITMSK: .RES   1          ;BIT MASK
006F                SHFAMT: .RES   1
0070                ROWAC:  .RES   2
0072                COLAC:  .RES   2
```

```
 280 0074                ENDPT:  .RES    2
 281 0076                DELTAR: .RES    1
 282 0077                DELTAC: .RES    2
 283 0079                ROWINC: .RES    1
 284 007A                COLINC: .RES    1
 285 007B                SWPFLG: .RES    1       ;NON-0 IF TXT AND REGULAR RAM IS SWAPPED
 286 007C                HOLDCH: .RES    1       ;CH IS MOVED HERE IN KGETCH BEFORE CNTL & SH
 287 007D                INSDAT: .RES    1
 288 007E                COUNTR: .RES    2
 289                     ;
 290                     ;
 291                     ;
 292                     ;
 293                     ;       80 - FF ARE RESERVED FOR USER APPLICATIONS
 294                     ;
 295                     ;
 296                     ;
 297                     ;       NOTE : SEE FLOATING POINT SUBROUTINE AREA FOR ZERO PAGE CELLS
 298                     ;
 299                     ;
 300                     ;
 301                     ;
 302                     ;       PAGE 1   -   STACK
 303                     ;
 304                     ;
 305                     ;
 306                     ;
 307                     ;       PAGE TWO RAM ASSIGNMENTS
 308                     ;
 309                             *=$0200
 310 0200                INTABS  =*                      ;INTERRUPT RAM
 311 0200                VDSLST: .RES    2               ;DISPLAY LIST NMI VECTOR
 312 0202                VPRCED: .RES    2               ;PROCEED LINE IRQ VECTOR
 313 0204                VINTER: .RES    2               ;INTERRUPT LINE IRQ VECTOR
 314 0206                VBREAK: .RES    2               ;SOFTWARE BREAK (00) INSTRUCTION IRQ VECTOR
 315 0208                VKEYBD: .RES    2               ;POKEY KEYBOARD IRQ VECTOR
 316 020A                VSERIN: .RES    2               ;POKEY SERIAL INPUT READY IRQ
 317 020C                VSEROR: .RES    2               ;POKEY SERIAL OUTPUT READY IRQ
 318 020E                VSEROC: .RES    2               ;POKEY SERIAL OUTPUT COMPLETE IRQ
 319 0210                VTIMR1: .RES    2               ;POKEY TIMER 1 IRQ
 320 0212                VTIMR2: .RES    2               ;POKEY TIMER 2 IRQ
 321 0214                VTIMR4: .RES    2               ;POKEY TIMER 4 IRQ
 322 0216                VIMIRQ: .RES    2               ;IMMEDIATE IRQ VECTOR
 323 0218                CDTMV1: .RES    2               ;COUNT DOWN TIMER 1
 324 021A                CDTMV2: .RES    2               ;COUNT DOWN TIMER2
 325 021C                CDTMV3: .RES    2               ;COUNT DOWN TIMER 3
 326 021E                CDTMV4: .RES    2               ;COUNT DOWN TIMER 4
 327 0220                CDTMV5: .RES    2               ;COUNT DOWN TIMER 5
 328 0222                VVBLKI: .RES    2               ;IMMEDIATE VERTICAL BLANK NMI VECTOR
 329 0224                VVBLKD: .RES    2               ;DEFERRED VERTICAL BLANK NMI VECTOR
 330 0226                CDTMA1: .RES    2               ;COUNT DOWN TIMER 1 JSR ADDRESS
 331 0228                CDTMA2: .RES    2               ;COUNT DOWN TIMER 2 JSR ADDRESS
 332 022A                CDTMF3: .RES    1               ;COUNT DOWN TIMER 3 FLAG
 333 022B                SRTIMR: .RES    1               ;SOFTWARE REPEAT TIMER
```

```
022C                CDTMF4:  RES    1          ;COUNT DOWN TIMER 4 FLAG
022D                INTEMP:  RES    1          ;IAN'S TEMP (RENAMED FROM T1 BY POPULAR DEMA
022E                CDTMF5:  RES    1          ;COUNT DOWN TIMER FLAG 5
022F                SDMCTL:  RES    1          ;SAVE DMACTL REGISTER
0230                SDLSTL:  RES    1          ;SAVE DISPLAY LIST LOW BYTE
0231                SDLSTH:  RES    1          ;SAVE DISPLAY LIST HI BYTE
0232                SSKCTL:  RES    1          ;SKCTL REGISTER RAM
0233                         RES    1          ;
                    ;
0234                LPENH:   RES    1          ;LIGHT PEN HORIZONTAL VALUE
0235                LPENV:   RES    1          ;LIGHT PEN VERTICAL VALUE
0236                BRKKY:   RES    2          ;BREAK KEY VECTOR
                    ;
0238                         RES    2          ;SPARE
                    ;
023A                CDEVIC:  RES    1          ;COMMAND FRAME BUFFER - DEVICE
023B                CCOMND:  RES    1          ;COMMAND
023C                CAUX1:   RES    1          ;COMMAND AUX BYTE 1
023D                CAUX2:   RES    1          ;COMMAND AUX BYTE 2
                    ;    NOTE: MAY NOT BE THE LAST WORD ON A PAGE
023E                TEMP:    RES    1          ;TEMPORARY RAM CELL
                    ;    NOTE: MAY NOT BE THE LAST WORD ON A PAGE
023F                ERRFLG:  RES    1          ;ERROR FLAG - ANY DEVICE ERROR EXCEPT TIME 0
                    ;
0240                DFLAGS:  RES    1          ;DISK FLAGS FROM SECTOR ONE
0241                DBSECT:  RES    1          ;NUMBER OF DISK BOOT SECTORS
0242                BOOTAD:  RES    2          ;ADDRESS WHERE DISK BOOT LOADER WILL BE PUT
0244                COLDST:  RES    1          ;COLDSTART FLAG (1=IN MIDDLE OF COLDSTART)
                    ;
0245                         RES    1          ;SPARE
                    ;
0246                DSKTIM:  RES    1          ;DISK TIME OUT REGISTER
                    ;
0247                LINBUF:  RES    40         ;CHAR LINE BUFFER
                    ;
026F                GPRIOR:  RES    1          ;GLOBAL PRIORITY CELL
                    ;
0270                PADDL0:  RES    1          ;POTENTIOMETER 0 RAM CELL
0271                PADDL1:  RES    1
0272                PADDL2:  RES    1
0273                PADDL3:  RES    1
0274                PADDL4:  RES    1
0275                PADDL5:  RES    1
0276                PADDL6:  RES    1
0277                PADDL7:  RES    1
0278                STICK0:  RES    1          ;JOYSTICK 0 RAM CELL
0279                STICK1:  RES    1
027A                STICK2:  RES    1
027B                STICK3:  RES    1
027C                PTRIG0:  RES    1          ;PADDLE TRIGGER 0
027D                PTRIG1:  RES    1
027E                PTRIG2:  RES    1
027F                PTRIG3:  RES    1
0280                PTRIG4:  RES    1
```

```
     388 0281                PTRIG5:  RES   1
     389 0282                PTRIG6:  RES   1
     390 0283                PTRIG7:  RES   1
     391 0284                STRIG0:  RES   1          ;JOYSTICK TRIGGER 0
     392 0285                STRIG1:  RES   1
     393 0286                STRIG2:  RES   1
     394 0287                STRIG3:  RES   1
     395                     ;
     396 0288                CSTAT:   RES   1
     397 0289                WMODE:   RES   1
     398 028A                BLIM:    RES   1
     399 028B                IMASK:   RES   1
     400 028C                JVECK:   RES   2
     401                     ;
     402 028E                         RES   2          ;SPARE
     403                     ;
     404                     ;
     405                     ;
     406                     ;
     407 0290                TXTROW:  RES   1          ;TEXT ROWCRS
     408 0291                TXTCOL:  RES   2          ;TEXT COLCRS
     409 0293                TINDEX:  RES   1          ;TEXT INDEX
     410 0294                TXTMSC:  RES   2          ;FOOLS CONVRT INTO NEW MSC
     411 0296                TXTOLD:  RES   6          ;OLDROW & OLDCOL FOR TEXT (AND THEN SOME)
     412 029C                TMPX1:   RES   1
     413 029D                HOLD3:   RES   1
     414 029E                SUBTMP:  RES   1
     415 029F                HOLD2:   RES   1
     416 02A0                DMASK:   RES   1
     417 02A1                TMPLBT:  RES   1
     418 02A2                ESCFLG:  RES   1          ;ESCAPE FLAG
     419 02A3                TABMAP:  RES   15
     420 02B2                LOGMAP:  RES   4          ;LOGICAL LINE START BIT MAP
     421 02B6                INVFLG:  RES   1          ;INVERSE VIDEO FLAG (TOGGLED BY ATARI KEY)
     422 02B7                FILFLG:  RES   1          ;RIGHT FILL FLAG FOR DRAW
     423 02B8                TMPROW:  RES   1
     424 02B9                TMPCOL:  RES   2
     425 02BB                SCRFLG:  RES   1          ;SET IF SCROLL OCCURS
     426 02BC                HOLD4:   RES   1          ;TEMP CELL USED IN DRAW ONLY
     427 02BD                HOLD5:   RES   1          ;DITTO
     428 02BE                SHFLOK:  RES   1
     429 02BF                BOTSCR:  RES   1          ;BOTTOM OF SCREEN : 24 NORM 4 SPLIT
     430                     ;
     431                     ;
     432 02C0                PCOLR0:  RES   1          ;P0 COLOR
     433 02C1                PCOLR1:  RES   1          ;P1 COLOR
     434 02C2                PCOLR2:  RES   1          ;P2 COLOR
     435 02C3                PCOLR3:  RES   1          ;P3 COLOR
     436 02C4                COLOR0:  RES   1          ;COLOR 0
     437 02C5                COLOR1:  RES   1
     438 02C6                COLOR2:  RES   1
     439 02C7                COLOR3:  RES   1
     440 02C8                COLOR4:  RES   1
     441                     ;
```

```
 442                            ;
 443  02C9                      .RES    23          ;SPARE
 444                            ;
 445                            ;
 446                            ;
 447  02E0                      GLBABS  =*          ;GLOBAL VARIABLES
 448                            ;
 449  02E0                      .RES    4           ;SPARE
 450                            ;
 451  02E4                      RAMSIZ: .RES    1   ;RAM SIZE (HI BYTE ONLY)
 452  02E5                      MEMTOP: .RES    2   ;TOP OF AVAILABLE USER MEMORY
 453  02E7                      MEMLO:  .RES    2   ;BOTTOM OF AVAILABLE USER MEMORY
 454  02E9                      .RES    1           ;SPARE
 455  02EA                      DVSTAT: .RES    4   ;STATUS BUFFER
 456  02EE                      CBAUDL: .RES    1   ;CASSETTE BAUD RATE LOW BYTE
 457  02EF                      CBAUDH: .RES    1
 458                            ;
 459  02F0                      CRSINH: .RES    1   ;CURSOR INHIBIT (00 = CURSOR ON)
 460  02F1                      KEYDEL: .RES    1   ;KEY DELAY
 461  02F2                      CH1:    .RES    1
 462                            ;
 463  02F3                      CHACT:  .RES    1   ;CHACTL REGISTER RAM
 464  02F4                      CHBAS:  .RES    1   ;CHBAS REGISTER RAM
 465                            ;
 466  02F5                      .RES    5           ;SPARE BYTES
 467                            ;
 468  02FA                      CHAR:   .RES    1
 469  02FB                      ATACHR: .RES    1   ;ATASCII CHARACTER
 470  02FC                      CH:     .RES    1   ;GLOBAL VARIABLE FOR KEYBOARD
 471  02FD                      FILDAT: .RES    1   ;RIGHT FILL DATA (DRAW)
 472  02FE                      DSPFLG: .RES    1   ;DISPLAY FLAG : DISPLAY CNTLS IF NON-ZERO
 473  02FF                      SSFLAG: .RES    1   ;START/STOP FLAG FOR PAGING (CNTL 1). CLEARE
 474                            ;
 475                            ;
 476                            ;
 477                            ;
 478                            ;
 479                            ;
 480                            ;
 481                            ;       PAGE THREE RAM ASSIGNMENTS
 482                            ;
 483  0300                      DCB     =*              ;DEVICE CONTROL BLOCK
 484  0300                      DDEVIC: .RES    1       ;PERIPHERAL UNIT 1 BUS I.D. NUMBER
 485  0301                      DUNIT:  .RES    1       ;UNIT NUMBER
 486  0302                      DCOMND: .RES    1       ;BUS COMMAND
 487  0303                      DSTATS: .RES    1       ;COMMAND TYPE/STATUS RETURN
 488  0304                      DBUFLO: .RES    1       ;DATA BUFFER POINTER LOW BYTE
 489  0305                      DBUFHI: .RES    1
 490  0306                      DTIMLO: .RES    1       ;DEVICE TIME OUT IN 1 SECOND UNITS
 491  0307                      DUNUSE: .RES    1       ;UNUSED BYTE
 492  0308                      DBYTLO: .RES    1       ;NUMBER OF BYTES TO BE TRANSFERRED LOW BYTE
 493  0309                      DBYTHI: .RES    1
 494  030A                      DAUX1:  .RES    1       ;COMMAND AUXILIARY BYTE 1
 495  030B                      DAUX2:  .RES    1
```

```
 496                    ;
 497  030C              TIMER1: .RES    2           ;INITIAL TIMER VALUE
 498  030E              ADDCOR: .RES    1           ;ADDITION CORRECTION
 499  030F              CASFLG: .RES    1           ;CASSETTE MODE WHEN SET
 500  0310              TIMER2: .RES    2           ;FINAL TIMER VALUE. THESE TWO TIMER VALUES
 501                    ; ARE USED TO COMPUTE INTERVAL FOR BAUD RATE
 502  0312              TEMP1:  .RES    2           ;TEMPORARY STORAGE REGISTER
 503  0314              TEMP2:  .RES    1           ;TEMPORARY STORAGE REGISTER
 504  0315              TEMP3:  .RES    1           ;TEMPORARY STORAGE REGISTER
 505  0316              SAVIO:  .RES    1           ;SAVE SERIAL IN DATA PORT
 506  0317              TIMFLG: .RES    1           ;TIME OUT FLAG FOR BAUD RATE CORRECTION
 507  0318              STACKP: .RES    1           ;SIO STACK POINTER SAVE CELL
 508  0319              TSTAT:  .RES    1           ;TEMPORARY STATUS HOLDER
 509                    ;
 510                    ;
 511                    ;
 512  031A              HATABS: .RES    38          ;HANDLER ADDRESS TABLE
 513  0021              MAXDEV  =       *-HATABS-5  ;MAXIMUM HANDLER ADDRESS INDEX
 514                    ;
 515                    ;       NOTE : THE ENTIRE IOCB DEFINITIONS HAVE BEEN MODIFIED
 516                    ;
 517                    IOCB:   .ORG    *           ;I/O CONTROL BLOCKS
 518  0340              ICHID:  .RES    1           ;HANDLER INDEX NUMBER (FF = IOCB FREE)
 519  0341              ICDNO:  .RES    1           ;DEVICE NUMBER (DRIVE NUMBER)
 520  0342              ICCOM:  .RES    1           ;COMMAND CODE
 521  0343              ICSTA:  .RES    1           ;STATUS OF LAST IOCB ACTION
 522  0344              ICBAL:  .RES    1           ;BUFFER ADDRESS LOW BYTE
 523  0345              ICBAH:  .RES    1
 524  0346              ICPTL:  .RES    1           ;PUT BYTE ROUTINE ADDRESS - 1
 525  0347              ICPTH:  .RES    1
 526  0348              ICBLL:  .RES    1           ;BUFFER LENGTH LOW BYTE
 527  0349              ICBLH:  .RES    1
 528  034A              ICAX1:  .RES    1           ;AUXILIARY INFORMATION FIRST BYTE
 529  0348              ICAX2:  .RES    1
 530  034C              ICSPR:  .RES    4           ;FOUR SPARE BYTES
 531  0350                      .RES    MAXIOC-IOCBSZ
 532                    ;
 533  03C0              PRNBUF: .RES    40          ;PRINTER BUFFER
 534                    ;
 535  03E8                      .RES    21          ;SPARE BYTES
 536                    ;
 537                    ;
 538                    ;
 539                    ;
 540                    ;
 541                    ;
 542                    ;
 543                    ;       PAGE FOUR RAM ASSIGNMENTS
 544                    ;
 545  03FD              CASBUF: .RES    131         ;CASSETTE BUFFER
 546                    ;
 547                    ; USER AREA STARTS HERE AND GOES TO END OF PAGE FIVE
 548  0480              USAREA: .RES    128         ;SPARE
 549                    ;
```

```
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;       PAGE FIVE RAM ASSIGNMENTS
                                ;
                                ;       PAGE FIVE IS RESERVED AS A USER WORK SPACE
                                ;
                                ;       NOTE: SEE FLOATING POINT SUBROUTINE AREA FOR PAGE FIVE CELLS
                                ;
                                ;
                                ;       PAGE SIX RAM ASSIGNMENTS
                                ;
                                ; PAGE SIX IS RESERVED AS A USER'S USER WORK SPACE
                                ;
                                ;
                                ;
                                ;
                                ;       FLOATING POINT SUBROUTINES
                                ;
0006                            FPREC   =       6               ;FLOATING PT PRECISION (# OF BYTES)
                                ; IF CARRY USED THEN CARRY CLEAR => NO ERROR, CARR
D800                            AFP     =       $D800           ;ASCII->FLOATING POINT (FP)
                                ;                                   INBUFF+CIX -> FR0, CIX, CARRY
D8E6                            FASC    =       $D8E6           ;FP -> ASCII FR0-> LBUFF (INBUFF)
D9AA                            IFP     =       $D9AA           ;INTEGER -> FP
                                ;                                   0-$FFFF (LSB,MSB) IN FR0,FR0+1->FR0
D9D2                            FPI     =       $D9D2           ;FP -> INTEGER FR0 -> FR0,FR0+1, CARRY
DA60                            FSUB    =       $DA60           ;FR0 <- FR0 - FR1 ,CARRY
DA66                            FADD    =       $DA66           ;FR0 <- FR0 + FR1 ,CARRY
DADB                            FMUL    =       $DADB           ;FR0 <- FR0 * FR1 ,CARRY
DB28                            FDIV    =       $DB28           ;FR0 <- FR0 / FR1 ,CARRY
DD89                            FLD0R   =       $DD89           ;FLOATING LOAD REG0   FR0  <- (X,Y)
DD8D                            FLD0P   =       $DD8D           ;    "       "    "     FR0  <- (FLPTR)
DD98                            FLD1R   =       $DD98           ;    "       "   REG1   FR1  <- (X,Y)
DD9C                            FLD1P   =       $DD9C           ;    "       "    "     FR1  <- (FLPTR)
DDA7                            FSTOR   =       $DDA7           ;FLOATING STORE REG0 (X,Y) <- FR0
DDAB                            FSTOP   =       $DDAB           ;     "      "    " (FLPTR)  <- FR0
DDB6                            FMOVE   =       $DDB6           ;FR1 <- FR0
DD40                            PLYEVL  =       $DD40           ;FR0 <- P(Z) = SUM(I=N TO 0) (A(I)*Z**I) CAR
                                ;                               INPUT:  (X,Y) = A(N),A(N-1)...A(0)  -> PLYARG
                                ;                                       ACC   = # OF COEFFICIENTS = DEGREE+1
                                ;                                       FR0   = Z
DDC0                            EXP     =       $DDC0           ;FR0 <- E**FR0 = EXP10(FR0 * LOG10(E)) CARRY
DDCC                            EXP10   =       $DDCC           ;FR0 <- 10**FR0 CARRY
DECD                            LOG     =       $DECD           ;FR0 <- LN(FR0) = LOG10(FR0)/LOG10(E) CARRY
DED1                            LOG10   =       $DED1           ;FR0 <- LOG10 (FR0) CARRY
                                ; THE FOLLOWING ARE IN BASIC CARTRIDGE:
BD81                            SIN     =       $BD81           ;FR0 <- SIN(FR0) DEGFLG=0 =>RADS, 6=>DEG. CA
BD73                            COS     =       $BD73           ;FR0 <- COS(FR0) CARRY
BE43                            ATAN    =       $BE43           ;FR0 <- ATAN(FR0) CARRY
BEB1                            SQR     =       $BEB1           ;FR0 <- SQUAREROOT(FR0) CARRY
```

```
 604                    ; FLOATING POINT ROUTINES ZERO PAGE (NEEDED ONLY IF F.P. ROUTINES ARE CA
 605                            *=$D4
 606  00D4              FRO:     .RES    FPREC          ;FP REG0
 607  00DA              FRE:     .RES    FPREC
 608  00E0              FR1:     .RES    FPREC          ;FP REG1
 609  00E6              FR2:     .RES    FPREC
 610  00EC              FRX:     .RES    1              ;FP SPARE
 611  00ED              EEXP:    .RES    1              ;VALUE OF E
 612  00EE              NSIGN:   .RES    1              ;SIGN OF #
 613  00EF              ESIGN:   .RES    1              ;SIGN OF EXPONENT
 614  00F0              FCHRFLG: .RES    1              ;1ST CHAR FLAG
 615  00F1              DIGRT:   .RES    1              ;# OF DIGITS RIGHT OF DECIMAL
 616  00F2              CIX:     .RES    1              ;CURRENT INPUT INDEX
 617  00F3              INBUFF:  .RES    2              ;POINTS TO USER'S LINE INPUT BUFFER
 618  00F5              ZTEMP1:  .RES    2
 619  00F7              ZTEMP4:  .RES    2
 620  00F9              ZTEMP3:  .RES    2
 621  00FB              DEGFLG
 622  00FB              RADFLG:  .RES    1              ;0=RADIANS, 6=DEGREES
 623  0000              RADON    =       0              ;INDICATES RADIANS
 624  0006              DEGON    =       6              ;INDICATES DEGREES
 625  00FC              FLPTR:   .RES    2              ;POINTS TO USER'S FLOATING PT NUMBER
 626  00FE              FPTR2:   .RES    2
 627                    ; FLOATING PT ROUTINES' NON-ZERO PAGE RAM
 628                    ; (NEEDED ONLY IF F.P. ROUTINES CALLED)
 629                            *=$57E
 630  057E              LBPR1:   .RES    1              ;LBUFF PREFIX 1
 631  057F              LBPR2:   .RES    1              ;LBUFF PREFIX 2
 632  0580              LBUFF:   .RES    128            ;LINE BUFFER
 633  05E0              PLYARG   =       LBUFF+$60      ;POLYNOMIAL ARGUMENTS
 634  05E6              FPSCR    =       PLYARG+FPREC
 635  05EC              FPSCR1   =       FPSCR+FPREC
 636  05E6              FSCR     =       FPSCR
 637  05EC              FSCR1    =       FPSCR1
 638  05FF              LBFEND   =       *-1            ;END OF LBUFF
 639                    ;
 640                    ;
 641                    ;
 642                    ;
 643                    ;
 644                    ;
 645                    ;
 646                    ;
 647                    ;
 648                    ;       COLLEEN MNEMONICS
 649                    ;
 650  D200              POKEY    =       $D200          ;VBLANK ACTION:          DESCRIPTION:
 651  D200              POT0     =       POKEY+0        ;POT0-->PADDL0           0-227 IN RAM CELL
 652  D201              POT1     =       POKEY+1        ;POT1-->PADDL1           0-227 IN RAM CELL
 653  D202              POT2     =       POKEY+2        ;POT2-->PADDL2           0-227 IN RAM CELL
 654  D203              POT3     =       POKEY+3        ;POT3-->PADDL3           0-227 IN RAM CELL
 655  D204              POT4     =       POKEY+4        ;POT4-->PADDL4           0-227 IN RAM CELL
 656  D205              POT5     =       POKEY+5        ;POT5-->PADDL5           0-227 IN RAM CELL
 657  D206              POT6     =       POKEY+6        ;POT6-->PADDL6           0-227 IN RAM CELL
```

```
     658 D207                POT7    =       POKEY+7     ;POT7-->PADDL7          0-227 IN RAM CELL
     659 D208                ALLPOT  =       POKEY+8     ;???
     660 D209                KBCODE  =       POKEY+9
     661 D20A                RANDOM  =       POKEY+10
     662 D20B                POTGO   =       POKEY+11    ;STROBED
     663 D20D                SERIN   =       POKEY+13
     664 D20E                IRQST   =       POKEY+14
     665 D20F                SKSTAT  =       POKEY+15
     666 D200                AUDF1   =       POKEY+0
     667 D201                AUDC1   =       POKEY+1
     668 D202                AUDF2   =       POKEY+2
     669 D203                AUDC2   =       POKEY+3
     670 D204                AUDF3   =       POKEY+4
     671 D205                AUDC3   =       POKEY+5
     672 D206                AUDF4   =       POKEY+6
     673 D207                AUDC4   =       POKEY+7
     674 D208                AUDCTL  =       POKEY+8     ;NONE                AUDCTL<--[SIO]
     675 D209                STIMER  =       POKEY+9
     676 D20A                SKRES   =       POKEY+10    ;NONE                 SKRES<--[SIO]
     677 D20D                SEROUT  =       POKEY+13    ;NONE                SEROUT<--[SIO]
     678 D20E                IRQEN   =       POKEY+14    ;POKMSK-->IRQEN (AFFECTED BY OPEN S: OR E:)
     679 D20F                SKCTL   =       POKEY+15    ;SSKCTL-->SKCTL      SSKCTL<--[SIO]
     680                     ;
     681 D000                CTIA    =       $D000       ;VBLANK ACTION:       DESCRIPTION:
     682 D000                HPOSP0  =       CTIA+0
     683 D001                HPOSP1  =       CTIA+1
     684 D002                HPOSP2  =       CTIA+2
     685 D003                HPOSP3  =       CTIA+3
     686 D004                HPOSM0  =       CTIA+4
     687 D005                HPOSM1  =       CTIA+5
     688 D006                HPOSM2  =       CTIA+6
     689 D007                HPOSM3  =       CTIA+7
     690 D008                SIZEP0  =       CTIA+8
     691 D009                SIZEP1  =       CTIA+9
     692 D00A                SIZEP2  =       CTIA+10
     693 D00B                SIZEP3  =       CTIA+11
     694 D00C                SIZEM   =       CTIA+12
     695 D00D                GRAFP0  =       CTIA+13
     696 D00E                GRAFP1  =       CTIA+14
     697 D00F                GRAFP2  =       CTIA+15
     698 D010                GRAFP3  =       CTIA+16
     699 D011                GRAFM   =       CTIA+17
     700 D012                COLPM0  =       CTIA+18     ;PCOLR0-->COLPM0      WITH ATTRACT MODE
     701 D013                COLPM1  =       CTIA+19     ;PCOLR1-->COLPM1      WITH ATTRACT MODE
     702 D014                COLPM2  =       CTIA+20     ;PCOLR2-->COLPM2      WITH ATTRACT MODE
     703 D015                COLPM3  =       CTIA+21     ;PCOLR3-->COLPM3      WITH ATTRACT MODE
     704 D016                COLPF0  =       CTIA+22     ;COLOR0-->COLPF0      WITH ATTRACT MODE
     705 D017                COLPF1  =       CTIA+23     ;COLOR1-->COLPF1      WITH ATTRACT MODE
     706 D018                COLPF2  =       CTIA+24     ;COLOR2-->COLPF2      WITH ATTRACT MODE
     707 D019                COLPF3  =       CTIA+25     ;COLOR3-->COLPF3      WITH ATTRACT MODE
     708 D01A                COLBK   =       CTIA+26     ;COLOR4-->COLBK       WITH ATTRACT MODE
     709 D01B                PRIOR   =       CTIA+27     ;(ON OPEN S: OR E:)   GPRIOR-->PRIOR
     710 D01C                VDELAY  =       CTIA+28
     711 D01D                GRACTL  =       CTIA+29
```

```
  712 D01E                HITCLR  =       CTIA+30
  713 D01F                CONSOL  =       CTIA+31     ;$08-->CONSOL       TURN OFF SPEAKER
  714 D000                M0PF    =       CTIA+0
  715 D001                M1PF    =       CTIA+1
  716 D002                M2PF    =       CTIA+2
  717 D003                M3PF    =       CTIA+3
  718 D004                P0PF    =       CTIA+4
  719 D005                P1PF    =       CTIA+5
  720 D006                P2PF    =       CTIA+6
  721 D007                P3PF    =       CTIA+7
  722 D008                M0PL    =       CTIA+8
  723 D009                M1PL    =       CTIA+9
  724 D00A                M2PL    =       CTIA+10
  725 D00B                M3PL    =       CTIA+11
  726 D00C                P0PL    =       CTIA+12
  727 D00D                P1PL    =       CTIA+13
  728 D00E                P2PL    =       CTIA+14
  729 D00F                P3PL    =       CTIA+15
  730 D010                TRIG0   =       CTIA+16     ;TRIG0-->STRIG0
  731 D011                TRIG1   =       CTIA+17     ;TRIG1-->STRIG1
  732 D012                TRIG2   =       CTIA+18     ;TRIG2-->STRIG2
  733 D013                TRIG3   =       CTIA+19     ;TRIG3-->STRIG3
  734                     ;
  735 D400                ANTIC   =       $D400       ;VBLANK ACTION      DESCRIPTION
  736 D400                DMACTL  =       ANTIC+0     ;DMACTL<--SDMCTL    ON OPEN S: OR E:
  737 D401                CHACTL  =       ANTIC+1     ;CHACTL<--CHACT     ON OPEN S: OR E:
  738 D402                DLISTL  =       ANTIC+2     ;DLISTL<--SDLSTL    ON OPEN S: OR E:
  739 D403                DLISTH  =       ANTIC+3     ;DLISTH<--SDLSTH    ON OPEN S: OR E:
  740 D404                HSCROL  =       ANTIC+4
  741 D405                VSCROL  =       ANTIC+5
  742 D407                PMBASE  =       ANTIC+7
  743 D409                CHBASE  =       ANTIC+9     ;CHBASE<--CHBAS     ON OPEN S: OR E:
  744 D40A                WSYNC   =       ANTIC+10
  745 D40B                VCOUNT  =       ANTIC+11
  746 D40C                PENH    =       ANTIC+12
  747 D40D                PENV    =       ANTIC+13
  748 D40E                NMIEN   =       ANTIC+14    ;NMIEN<--40 POWER   ON AND [SETVBV]
  749 D40F                NMIRES  =       ANTIC+15    ;STROBED
  750 D40F                NMIST   =       ANTIC+15
  751 D300                PIA     =       $D300       ;VBLANK ACTION      DESCRIPTION
  752 D300                PORTA   =       PIA+0       ;PORTA-->STICK0,1   X-Y CONTROLLERS
  753 D301                PORTB   =       PIA+1       ;PORTB-->STICK2,3   X-Y CONTROLLERS
  754 D302                PACTL   =       PIA+2       ;NONE               PACTL<--3C [INIT]
  755 D303                PBCTL   =       PIA+3       ;NONE               PBCTL<--3C [INIT]
  756                     ;
  757                     ;
  758                     ;
  759                     ;  .PAGE
```

```
     760                                .PAGE
     761                                LIST    S
     762                                .TITLE  'CENTRAL INPUT/OUTPUT (CIO) 2-7-79'
     763                           ;            UPDATED BY AL MILLER 3-9-79
     764  0030                     ASCZER  =       '0          ;ASCII ZERO
     765  003A                     COLON   =       $3A         ;ASCII COLON
     766  009B                     EOL     =       $9B         ;END OF RECORD
```

```
                                .PAGE
                        ;
                        ; CIO JUMP VECTOR FOR USERS
                                *=CIOV
E456 4C C4 E4                   JMP     CIO             ;GO TO CIO
                        ;
                        ; CIO INIT JUMP VECTOR FOR POWER UP
                                *=CIOINV
E46E 4C A6 E4                   JMP     CIOINT          ;GO TO INIT
                        ;
                        ;
                        ; ERROR ROUTINE ADDRESS EQUATE
                        ; ERRTNH =ERRTN/256            "MOVED TO LINE 788"
                        ; ERRTNL =-ERRTNH*256+ERRTN    "MOVED TO LINE 789"
                        ;
                        ;
                                *=CIOORG
                        ;
                        ; CIO INITIALIZATION (CALLED BY MONITOR AT POWER UP)
E4A6 A2 00              CIOINT: LDX     #0
E4A8 A9 FF              CIOI1:  LDA     #IOCFRE         ;SET ALL IOCB'S TO FREE
E4AA 9D 40 03                   STA     ICHID,X         ;BY SETTING HANDLER ID'S=$FF
E4AD A9 C0                      LDA     #ERRTNL
E4AF 9D 46 03                   STA     ICPTL,X         ;POINT PUT TO ERROR ROUTINE
E4B2 A9 E4                      LDA     #ERRTNH
E4B4 9D 47 03                   STA     ICPTH,X
E4B7 8A                         TXA
E4B8 18                         CLC
E4B9 69 10                      ADC     #IOCBSZ         ;BUMP INDEX BY SIZE
E4BB AA                         TAX
E4BC C9 80                      CMP     #MAXIOC         ;DONE?
E4BE 90 E8                      BCC     CIOI1           ;NO
E4C0 60                         RTS                     ;YES, RETURN
                        ;
                        ; ERROR ROUTINE FOR ILLEGAL PUT
E4C0                    ERRTN   =*-1
00E4                    ERRTNH  =ERRTN/256
00C0                    ERRTNL  =(-ERRTNH)*256+ERRTN
E4C1 A0 85                      LDY     #NOTOPN         ;IOCB NOT OPEN
E4C3 60                         RTS
```

```
                        .PAGE
                ;
                ; CIO LOCAL RAM (USES SPARE BYTES IN ZERO PAGE IOCB)
002C            ENTVEC  =       ICSPRZ
                ;
                ; CIO MAIN ROUTINE
                ;
                ; CIO INTERFACES BETWEEN USER AND INPUT/OUTPUT DE
E4C4 85 2F      CIO:    STA     CIOCHR          ;SAVE POSSIBLE OUTPUT CHARACTER
E4C6 86 2E              STX     ICIDNO          ;SAVE IOCB NUMBER * N
                ;
                ; CHECK FOR LEGAL IOCB
E4C8 8A                 TXA
E4C9 29 0F              AND     #$F             ;IS IOCB MULTIPLE OF 16?
E4CB D0 04              BNE     CIERR1          ;NO, ERROR
E4CD E0 80              CPX     #MAXIOC         ;IS INDEX TOO LARGE?
E4CF 90 05              BCC     IOC1            ;NO
                ;
                ; INVALID IOCB NUMBER -- RETURN ERROR
E4D1 A0 86      CIERR1: LDY     #BADIOC         ;ERROR CODE
E4D3 4C 1B E6           JMP     CIRTN1          ;RETURN
                ;
                ; MOVE USER IOCB TO ZERO PAGE
E4D6 A0 00      IOC1:   LDY     #0
E4D8 BD 40 03   IOC1A:  LDA     IOCB,X          ;USER IOCB
E4DB 99 20 00           STA     IOCBAS,Y        ;TO ZERO PAGE
E4DE E8                 INX
E4DF C8                 INY
E4E0 C0 0C              CPY     #12             ;12 BYTES
E4E2 90 F4              BCC     IOC1A
                ;
                ; COMPUTE CIO INTERNAL VECTOR FOR COMMAND
E4E4 A0 84              LDY     #NVALID         ;ASSUME INVALID CODE
E4E6 A5 22              LDA     ICCOMZ          ;COMMAND CODE TO INDEX
E4E8 C9 03              CMP     #OPEN           ;IS COMMAND LEGAL?
E4EA 90 25              BCC     CIERR4          ;NO
E4EC A8                 TAY
                ;
                ; MOVE COMMAND TO ZERO BASE FOR INDEX
E4ED C0 0E              CPY     #SPECIL         ;IS COMMAND SPECIAL?
E4EF 90 02              BCC     IOC2            ;NO
E4F1 A0 0E              LDY     #SPECIL         ;YES, SET SPECIAL OFFSET INDEX
E4F3 84 17      IOC2:   STY     ICCOMT          ;SAVE COMMAND FOR VECTOR
E4F5 B9 C6 E6           LDA     COMTAB-3,Y      ;GET VECTOR OFFSET FROM TABLE
E4F8 F0 0F              BEQ     CIOPEN          ;GO IF OPEN COMMAND
E4FA C9 02              CMP     #2              ;IS IT CLOSE?
E4FC F0 35              BEQ     CICLOS          ;YES
E4FE C9 08              CMP     #8              ;IS IT STATUS OR SPECIAL?
E500 B0 4C              BCS     CISTSP          ;YES
E502 C9 04              CMP     #4              ;IS IT READ?
E504 F0 63              BEQ     CIREAD          ;YES
E506 4C C9 E5           JMP     CIWRIT          ;ELSE, MUST BE WRITE
```

```
                                        .PAGE
                                ;
                                ; OPEN COMMAND
                                ;
                                ; FIND DEVICE HANDLER IN HANDLER ADDRESS TABLE
E509  A5 20                     CIOPEN: LDA     ICHIDZ      ;GET HANDLER ID
E50B  C9 FF                             CMP     #IOCFRE     ;IS THIS IOCB CLOSED?
E50D  F0 05                             BEQ     IOC6        ;YES
                                ;
                                ; ERROR -- IOCB ALREADY OPEN
E50F  A0 81                     CIERR3: LDY     #PRVOPN     ;ERROR CODE
E511  4C 1B E6                  CIERR4: JMP     CIRTN1      ;RETURN
                                ;
                                ; GO FIND DEVICE
E514  20 9E E6                  IOC6:   JSR     DEVSRC      ;CALL DEVICE SEARCH
E517  B0 F8                             BCS     CIERR4      ;GO IF DEVICE NOT FOUND
                                ;
                                ; DEVICE FOUND, INITIALIZE IOCB FOR OPEN
                                ;
                                ; COMPUTE HANDLER ENTRY POINT
E519  20 3D E6                  IOC7:   JSR     COMENT
E51C  B0 F3                             BCS     CIERR4      ;GO IF ERROR IN COMPUTE
                                ;
                                ; GO TO HANDLER FOR INITIALIZATION
E51E  20 89 E6                          JSR     GOHAND      ;USE INDIRECT JUMP
                                ;
                                ; STORE PUT BYTE ADDRESS-1 INTO IOCB
E521  A9 0B                             LDA     #PUTCHR     ;SIMULATE PUT CHARACTER
E523  85 17                             STA     ICCOMT
E525  20 3D E6                          JSR     COMENT      ;COMPUTE ENTRY POINT
E528  A5 2C                             LDA     ICSPRZ      ;MOVE COMPUTED VALUE
E52A  85 26                             STA     ICPTLZ      ;TO PUT BYTE ADDRESS
E52C  A5 2D                             LDA     ICSPRZ+1
E52E  85 27                             STA     ICPTHZ
E530  4C 1D E6                          JMP     CIRTN2      ;RETURN TO USER
```

```
                                .PAGE
                        ;
                        ;
                        ; CLOSE COMMAND
E533 A0 01              CICLOS: LDY     #SUCCES     ;ASSUME GOOD CLOSE
E535 84 23                      STY     ICSTAZ
E537 20 3D E6                   JSR     COMENT      ;COMPUTE HANDLER ENTRY POINT
E53A B0 03                      BCS     CICLO2      ;GO IF ERROR IN COMPUTE
E53C 20 89 E6                   JSR     GOHAND      ;GO TO HANDLER TO CLOSE DEVICE
E53F A9 FF              CICLO2: LDA     #IOCFRE     ;GET IOCB "FREE" VALUE
E541 85 20                      STA     ICHIDZ      ;SET HANDLER ID
E543 A9 E4                      LDA     #ERRTNH
E545 85 27                      STA     ICPTHZ      ;SET PUT BYTE TO POINT TO ERROR
E547 A9 C0                      LDA     #ERRTNL
E549 85 26                      STA     ICPTLZ
E54B 4C 1D E6                   JMP     CIRTN2      ;RETURN
                        ;
                        ;
                        ; STATUS AND SPECIAL REQUESTS
                        ; DO IMPLIED OPEN IF NECESSARY AND GO TO DEVICE
E54E A5 20              CISTSP: LDA     ICHIDZ      ;IS THERE A HANDLER ID?
E550 C9 FF                      CMP     #IOCFRE
E552 D0 05                      BNE     CIST1       ;YES
                        ;
                        ; IOCB IS FREE, DO IMPLIED OPEN
E554 20 9E E6                   JSR     DEVSRC      ;FIND DEVICE IN TABLE
E557 B0 B8                      BCS     CIERR4      ;GO IF ERROR IN COMPUTE
                        ;
                        ; COMPUTE AND GO TO ENTRY POINT IN HANDLER
E559 20 3D E6           CIST1:  JSR     COMENT      ;COMPUTER HANDLER ENTRY VECTOR
E55C 20 89 E6                   JSR     GOHAND      ;GO TO HANDLER
                        ;
                        ; RESTORE HANDLER INDEX (DO IMPLIED CLOSE)
E55F A6 2E                      LDX     ICIDNO      ;IOCB INDEX
E561 BD 40 03                   LDA     ICHID,X     ;GET ORIGINAL HANDLER ID
E564 85 20                      STA     ICHIDZ      ;RESTORE ZERO PAGE
E566 4C 1D E6                   JMP     CIRTN2      ;RETURN
```

```
                                .PAGE
                        ;
                        ; READ -- DO GET COMMANDS
E569 A5 22              CIREAD: LDA     ICCOMZ          ;GET COMMAND BYTE
E56B 25 2A                      AND     ICAX1Z          ;IS THIS READ LEGAL?
E56D D0 05                      BNE     RCI1A           ;YES
                        ;
                        ; ILLEGAL READ -- IOCB OPENED FOR WRITE ONLY
E56F A0 83                      LDY     #WRONLY         ;ERROR CODE
E571 4C 1B E6           RCI1B:  JMP     CIRTN1          ;RETURN
                        ;
                        ; COMPUTE AND CHECK ENTRY POINT
E574 20 3D E6           RCI1A:  JSR     COMENT          ;COMPUTE ENTRY POINT
E577 B0 F8                      BCS     RCI1B           ;GO IF ERROR IN COMPUTE
                        ;
                        ; GET RECORD OR CHARACTERS
E579 A5 28                      LDA     ICBLLZ
E57B 05 29                      ORA     ICBLLZ+1        ;IS BUFFER LENGTH ZERO?
E57D D0 08                      BNE     RCI3            ;NO
E57F 20 89 E6                   JSR     GOHAND
E582 85 2F                      STA     CIOCHR
E584 4C 1D E6                   JMP     CIRTN2
                        ;
                        ; LOOP TO FILL BUFFER OR END RECORD
E587 20 89 E6           RCI3:   JSR     GOHAND          ;GO TO HANDLER TO GET BYTE
E58A 85 2F                      STA     CIOCHR          ;SAVE BYTE
E58C 30 35                      BMI     RCI4            ;END TRANSFER IF ERROR
E58E A0 00                      LDY     #0
E590 91 24                      STA     (ICBALZ),Y      ;PUT BYTE IN USER BUFFER
E592 20 70 E6                   JSR     INCBFP          ;INCREMENT BUFFER POINTER
E595 A5 22                      LDA     ICCOMZ          ;GET COMMAND CODE
E597 29 02                      AND     #2              ;IS IT GET RECORD?
E599 D0 0C                      BNE     RCI1            ;NO
                        ;
                        ; CHECK FOR EOL ON TEXT RECORDS
E59B A5 2F                      LDA     CIOCHR          ;GET BYTE
E59D C9 9B                      CMP     #EOL            ;IS IT AN EOL?
E59F D0 06                      BNE     RCI1            ;NO
E5A1 20 63 E6                   JSR     DECBFL          ;YES, DECREMENT BUFFER LENGTH
E5A4 4C C3 E5                   JMP     RCI4            ;END TRANSFER
                        ;
                        ; CHECK BUFFER FULL
E5A7 20 63 E6           RCI1:   JSR     DECBFL          ;DECREMENT BUFFER LENGTH
E5AA D0 DB                      BNE     RCI3            ;CONTINUE IF NON ZERO
```

```
 975                            .PAGE
 976                    ;
 977                    ; BUFFER FULL, RECORD NOT ENDED
 978                    ; DISCARD BYTES UNTIL END OF RECORD
 979  E5AC  A5 22       RCI2:   LDA     ICCOMZ          ;GET COMMAND BYTE
 980  E5AE  29 02             AND     #2              ;IS IT GET CHARACTER?
 981  E5B0  D0 11             BNE     RCI4            ;YES, END TRANSFER
 982                    ;
 983                    ; LOOP TO WAIT FOR EOL
 984  E5B2  20 89 E6    RCI6:   JSR     GOHAND          ;GET BYTE FROM HANDLER
 985  E5B5  85 2F             STA     CIOCHR          ;SAVE CHARACTER
 986  E5B7  30 0A             BMI     RCI4            ;GO IF ERROR
 987                    ;
 988                    ; TEXT RECORD, WAIT FOR EOL
 989  E5B9  A5 2F             LDA     CIOCHR          ;GET GOT BYTE
 990  E5BB  C9 9B             CMP     #EOL            ;IS IT EOL?
 991  E5BD  D0 F3             BNE     RCI6            ;NO, CONTINUE
 992                    ;
 993                    ; END OF RECORD, BUFFER FULL -- SEND TRUNCATED RECORD MESSAGE
 994  E5BF  A9 89       RCI11:  LDA     #TRNRCD         ;ERROR CODE
 995  E5C1  85 23             STA     ICSTAZ          ;STORE IN IOCB
 996                    ;
 997                    ; TRANSFER DONE
 998  E5C3  20 77 E6    RCI4:   JSR     SUBBFL          ;SET FINAL BUFFER LENGTH
 999  E5C6  4C 1D E6          JMP     CIRTN2          ;RETURN
```

```
1000                            .PAGE
1001                    ;
1002                    ; WRITE -- DO PUT COMMANDS
1003  E5C9  A5 22       CIWRIT: LDA     ICCOMZ      ;GET COMMAND BYTE
1004  E5CB  25 2A               AND     ICAX1Z      ;IS THIS WRITE LEGAL?
1005  E5CD  D0 05               BNE     WCI1A       ;YES
1006                    ;
1007                    ; ILLEGAL WRITE -- DEVICE OPENED FOR READ ONLY
1008  E5CF  A0 87               LDY     #RDONLY     ;ERROR CODE
1009  E5D1  4C 1B E6    WCI1B:  JMP     CIRTN1      ;RETURN
1010                    ;
1011                    ; COMPUTE AND CHECK ENTRY POINT
1012  E5D4  20 3D E6    WCI1A:  JSR     COMENT      ;COMPUTE HANDLER ENTRY POINT
1013  E5D7  B0 F8               BCS     WCI1B       ;GO IF ERROR IN COMPUTE
1014                    ;
1015                    ; PUT RECORD OR CHARACTERS
1016  E5D9  A5 28               LDA     ICBLLZ
1017  E5DB  05 29               ORA     ICBLLZ+1    ;IS BUFFER LENGTH ZERO?
1018  E5DD  D0 06               BNE     WCI3        ;NO
1019  E5DF  A5 2F               LDA     CIOCHR      ;GET CHARACTER
1020  E5E1  E6 28               INC     ICBLLZ      ;SET BUFFER LENGTH=1
1021  E5E3  D0 06               BNE     WCI4        ;THEN JUST TRANSFER ONE BYTE
1022                    ;
1023                    ; LOOP TO TRANSFER BYTES FROM BUFFER TO HANDLER
1024  E5E5  A0 00       WCI3:   LDY     #0
1025  E5E7  B1 24               LDA     (ICBALZ),Y  ;GET BYTE FROM BUFFER
1026  E5E9  85 2F               STA     CIOCHR      ;SAVE
1027  E5EB  20 89 E6    WCI4:   JSR     GOHAND      ;GO PUT BYTE
1028  E5EE  30 25               BMI     WCI5        ;END IF ERROR
1029  E5F0  20 70 E6            JSR     INCBFP      ;INCREMENT BUFFER POINTER
1030                    ;
1031                    ; CHECK FOR TEXT RECORD
1032  E5F3  A5 22               LDA     ICCOMZ      ;GET COMMAND BYTE
1033  E5F5  29 02               AND     #2          ;IS IT PUT RECORD?
1034  E5F7  D0 0C               BNE     WCI1        ;NO
1035                    ;
1036                    ; TEXT RECORD -- CHECK FOR EOL TRANSFER
1037  E5F9  A5 2F               LDA     CIOCHR      ;GET LAST CHARACTER
1038  E5FB  C9 9B               CMP     #EOL        ;IS IT AN EOL?
1039  E5FD  D0 06               BNE     WCI1        ;NO
1040  E5FF  20 63 E6            JSR     DECBFL      ;DECREMENT BUFFER LENGTH
1041  E602  4C 15 E6            JMP     WCI5        ;END TRANSFER
1042                    ;
1043                    ; CHECK FOR BUFFER EMPTY
1044  E605  20 63 E6    WCI1:   JSR     DECBFL      ;DECREMENT BUFFER LENGTH
1045  E608  D0 DB               BNE     WCI3        ;CONTINUE IF NON ZERO
```

```
1046                            .PAGE
1047                    ;
1048                    ; BUFFER EMPTY, RECORD NOT FILLED
1049                    ; CHECK TYPE OF TRANSFER
1050 E60A A5 22         WCI2:   LDA     ICCOMZ          ;GET COMMAND CODE
1051 E60C 29 02                 AND     #2              ;IS IT PUT CHARACTER?
1052 E60E D0 05                 BNE     WCI5            ;YES, END TRANSFER
1053                    ;
1054                    ; PUT RECORD (TEXT), BUFFER EMPTY, SEND EOL
1055 E610 A9 9B                 LDA     #EOL
1056 E612 20 89 E6              JSR     GOHAND          ;GO TO HANDLER
1057                    ;
1058                    ; END PUT TRANSFER
1059 E615 20 77 E6      WCI5:   JSR     SUBBFL          ;SET ACTUAL PUT BUFFER LENGTH
1060 E618 4C 1D E6              JMP     CIRTN2          ;RETURN
```

```
1061                            .PAGE
1062                    ;
1063                    ; CIO RETURNS
1064                    ; RETURNS WITH Y=STATUS
1065  E61B  84 23       CIRTN1: STY     ICSTAZ          ;SAVE STATUS
1066                    ;
1067                    ; RETURNS WITH STATUS STORED IN ICSTAZ
1068                    ; MOVE IOCB IN ZERO PAGE BACK TO USER AREA
1069  E61D  A4 2E       CIRTN2: LDY     ICIDNO          ;GET IOCB INDEX
1070  E61F  B9 44 03          LDA     ICBAL,Y
1071  E622  85 24             STA     ICBALZ          ;RESTORE USER BUFFER POINTER
1072  E624  B9 45 03          LDA     ICBAH,Y
1073  E627  85 25             STA     ICBAHZ
1074  E629  A2 00             LDX     #0              ;LOOP COUNT AND INDEX
1075  E62B  B5 20     CIRT3:  LDA     IOCBAS,X        ;ZERO PAGE
1076  E62D  99 40 03          STA     IOCB,Y          ;TO USER AREA
1077  E630  E8                INX
1078  E631  C8                INY
1079  E632  E0 0C             CPX     #12             ;12 BYTES
1080  E634  90 F5             BCC     CIRT3
1081                    ;
1082                    ; RESTORE A,X, & Y
1083  E636  A5 2F             LDA     CIOCHR          ;GET LAST CHARACTER
1084  E638  A6 2E             LDX     ICIDNO          ;IOCB INDEX
1085  E63A  A4 23             LDY     ICSTAZ          ;GET STATUS AND SET FLAGS
1086  E63C  60                RTS                     ;RETURN TO USER
```

```
1087                                    .PAGE
1088                            ;
1089                            ;
1090                            ; CIO SUBROUTINES
1091                            ;
1092                            ; COMENT -- CHECK AND COMPUTE HANDLER ENTRY POINT
1093  E63D  A4 20               COMENT: LDY     ICHIDZ          ;GET HANDLER INDEX
1094  E63F  C0 22                     CPY     #MAXDEV+1       ;IS IT A LEGAL INDEX?
1095  E641  90 04                     BCC     COM1            ;YES
1096                            ;
1097                            ; ILLEGAL HANDLER INDEX MEANS DEVICE NOT OPEN FOR OPERATION
1098  E643  A0 85                     LDY     #NOTOPN         ;ERROR CODE
1099  E645  B0 1B                     BCS     COM2            ;RETURN
1100                            ;
1101                            ; USE HANDLER ADDRESS TABLE AND COMMAND TABLE TO GET VECTOR
1102  E647  B9 1B 03            COM1:   LDA     HATABS+1,Y      ;GET LOW BYTE OF ADDRESS
1103  E64A  85 2C                     STA     ICSPRZ          ;AND SAVE IN POINTER
1104  E64C  B9 1C 03                  LDA     HATABS+2,Y      ;GET HI BYTE OF ADDRESS
1105  E64F  85 2D                     STA     ICSPRZ+1
1106  E651  A4 17                     LDY     ICCOMT          ;GET COMMAND CODE
1107  E653  B9 C6 E6                  LDA     COMTAB-3,Y      ;GET COMMAND OFFSET
1108  E656  A8                        TAY
1109  E657  B1 2C                     LDA     (ICSPRZ),Y      ;GET LOW BYTE OF VECTOR FROM
1110  E659  AA                        TAX                     ;HANDLER ITSELF AND SAVE
1111  E65A  C8                        INY
1112  E65B  B1 2C                     LDA     (ICSPRZ),Y      ;GET HI BYTE OF VECTOR
1113  E65D  85 2D                     STA     ICSPRZ+1
1114  E65F  86 2C                     STX     ICSPRZ          ;SET LO BYTE
1115  E661  18                        CLC                     ;SHOW NO ERROR
1116  E662  60                  COM2:   RTS
1117                            ;
1118                            ;
1119                            ; DECBFL -- DECREMENT BUFFER LENGTH DOUBLE BYTE
1120                            ; Z FLAG = 0 ON RETURN IF LENGTH = 0 AFTER DECREMENT
1121  E663  C6 28               DECBFL: DEC     ICBLLZ          ;DECREMENT LOW BYTE
1122  E665  A5 28                     LDA     ICBLLZ          ;CHECK IT
1123  E667  C9 FF                     CMP     #$FF            ;DID IT GO BELOW?
1124  E669  D0 02                     BNE     DECBF1          ;NO
1125  E66B  C6 29                     DEC     ICBLLZ+1        ;DECREMENT HI BYTE
1126  E66D  05 29               DECBF1: ORA     ICBLLZ+1        ;SET Z IF BOTH ARE ZERO
1127  E66F  60                        RTS
1128                            ;
1129                            ;
1130                            ; INCBFP -- INCREMENT WORKING BUFFER POINTER
1131  E670  E6 24               INCBFP: INC     ICBALZ          ;BUMP LOW BYTE
1132  E672  D0 02                     BNE     INCBF1          ;GO IF NOT ZERO
1133  E674  E6 25                     INC     ICBALZ+1        ;ELSE, BUMP HI BYTE
1134  E676  60                  INCBF1: RTS
1135                            ;
1136                            ;
1137                            ; SUBBFL -- SET BUFFER LENGTH = BUFFER LENGTH - WORKING BYTE COUNT
1138  E677  A6 2E               SUBBFL: LDX     ICIDNO          ;GET IOCB INDEX
1139  E679  38                        SEC
1140  E67A  BD 48 03                  LDA     ICBLL,X         ;GET LOW BYTE OF INITIAL LENGTH
```

```
1141  E67D  E5 28          SBC     ICBLLZ      ;SUBTRACT FINAL LOW BYTE
1142  E67F  85 28          STA     ICBLLZ      ;AND SAVE BACK
1143  E681  BD 49 03       LDA     ICBLH,X     ;GET HI BYTE
1144  E684  E5 29          SBC     ICBLLZ+1
1145  E686  85 29          STA     ICBLHZ
1146  E688  60             RTS
1147                       ;
1148                       ;
1149                       ; GOHAND -- GO INDIRECT TO A DEVICE HANDLER
1150                       ; Y= STATUS ON RETURN, N FLAG=1 IF ERROR ON RETURN
1151  E689  A0 92          GOHAND: LDY     #FNCNOT     ;PREPARE NO FUNCTION STATUS FOR HANDLER RTS
1152  E68B  20 93 E6       JSR     CIJUMP      ;USE THE INDIRECT JUMP
1153  E68E  84 23          STY     ICSTAZ      ;SAVE STATUS
1154  E690  C0 00          CPY     #0          ;AND SET N FLAG
1155  E692  60             RTS
1156                       ;
1157                       ; INDIRECT JUMP TO HANDLER BY PAUL'S METHOD
1158  E693  AA             CIJUMP: TAX             ;SAVE A
1159  E694  A5 2D          LDA     ICSPRZ+1    ;GET JUMP ADDRESS HI BYTE
1160  E696  48             PHA                 ;PUT ON STACK
1161  E697  A5 2C          LDA     ICSPRZ      ;GET JUMP ADDRESS LO BYTE
1162  E699  48             PHA                 ;PUT ON STACK
1163  E69A  8A             TXA                 ;RESTORE A
1164  E69B  A6 2E          LDX     ICIDNO      ;GET IOCB INDEX
1165  E69D  60             RTS                 ;GO TO HANDLER INDIRECTLY
```

```
1166                            .PAGE
1167                    ;
1168                    ; DEVSRC -- DEVICE SEARCH, FIND DEVICE IN HANDLER ADDRESS TABLE
1169                    ;
1170                    ; LOOP TO FIND DEVICE
1171  E69E  A0 00       DEVSRC: LDY     #0
1172  E6A0  B1 24               LDA     (ICBALZ),Y  ;GET DEVICE NAME FROM USER
1173  E6A2  F0 0C               BEQ     CIERR2
1174  E6A4  A0 21               LDY     #MAXDEV     ;INITIAL COMPARE INDEX
1175  E6A6  D9 1A 03    DEVS1:  CMP     HATABS,Y    ;IS THIS THE DEVICE?
1176  E6A9  F0 0A               BEQ     DEVS2       ;YES
1177  E6AB  88                  DEY
1178  E6AC  88                  DEY                 ;ELSE, POINT TO NEXT DEVICE NAME
1179  E6AD  88                  DEY
1180  E6AE  10 F6               BPL     DEVS1       ;CONTINUE FOR ALL DEVICES
1181                    ;
1182                    ; NO DEVICE FOUND, DECLARE NON-EXISTENT DEVICE ERROR
1183  E6B0  A0 82       CIERR2: LDY     #NONDEV     ;ERROR CODE
1184  E6B2  38                  SEC                 ;SHOW ERROR
1185  E6B3  B0 13               BCS     DEVS4       ;AND RETURN
1186                    ;
1187                    ; FOUND DEVICE, SET ICHID,ICDNO, AND INIT DEVICE
1188  E6B5  98          DEVS2:  TYA
1189  E6B6  85 20               STA     ICHIDZ      ;SAVE HANDLER INDEX
1190  E6B8  38                  SEC
1191  E6B9  A0 01               LDY     #1
1192  E6BB  B1 24               LDA     (ICBALZ),Y  ;GET DEVICE NUMBER (DRIVE NUMBER)
1193  E6BD  E9 30               SBC     #ASCZER     ;SUBTRACT ASCII ZERO
1194  E6BF  C9 0A               CMP     #$A         ;IS NUMBER IN RANGE?
1195  E6C1  90 02               BCC     DEVS3       ;YES
1196  E6C3  A9 01               LDA     #1          ;NO, DEFAULT TO ONE
1197  E6C5  85 21       DEVS3:  STA     ICDNOZ      ;SAVE DEVICE NUMBER
1198  E6C7  18                  CLC                 ;SHOW NO ERROR
1199                    ;
1200                    ; RETURN
1201  E6C8  60          DEVS4:  RTS
```

```
1202                            .PAGE
1203
1204
1205                            ; CIO ROM TABLES
1206                            ;
1207                            ; COMMAND TABLE
1208                            ; MAPS EACH COMMAND TO OFFSET FOR APPROPRIATE VECTOR IN HANDLER
1209  E6C9  00 04 04 04         COMTAB: .BYTE   0,4,4,4,4,6,6,6,6,2,8,10
1210  E6CD  04 06 06 06
1211  E6D1  06 02 08 0A
1212  022F                    LENGTH  =*-CIOINT
1213  E6D5                    CRNTP1  =*
1214                                  *=$14
1215  0014  00                CIOSPR: .BYTE   INTORG-CRNTP1 ;^GCIOL IS TOO LONG
```

```
1216
1217                                    .TITLE  'INTERRUPT HANDLER'
1218                                ;LIVES ON DK1:INTHV.SRC
1219  0006                          SRTIM2  =       6               ;SECOND REPEAT INTERVAL
1220                                ;
1221                                ; THIS IS TO MAKE DOS 2 WORK WHICH USED AN ABSOLUTE ADDRESS
1222                                ;
1223                                        *=$E912
1224  E912  4C ED E8                       JMP     SETVBL
1225                                        *=SETVBV
1226  E45C  4C ED E8                       JMP     SETVBL
1227  E45F  4C AE E7                       JMP     SYSVBL
1228  E462  4C 05 E9                       JMP     XITVBL
1229                                        *=INTINV
1230  E46B  4C D5 E6                       JMP     IHINIT
1231                                ;
1232                                        *=VCTABL+INTABS-VDSLST
1233                                ;
1234  E480  90 E7                           .WORD   SYRTI           ;VDSLST
1235  E482  8F E7                           .WORD   SYIRQB          ;VPRCED
1236  E484  8F E7                           .WORD   SYIRQB          ;VINTER
1237  E486  8F E7                           .WORD   SYIRQB          ;VBREAK
1238                                ;
1239  E488                                  .RES    8
1240  E490  8F E7                           .WORD   SYIRQB          ;VTIMR1
1241  E492  8F E7                           .WORD   SYIRQB          ;VTIMR2
1242  E494  8F E7                           .WORD   SYIRQB          ;VTIMR4
1243  E496  06 E7                           .WORD   SYIRQ           ;VIMIRQ
1244  E498  00 00 00 00                     .WORD   0,0,0,0,0       ;CDTMV1-4
1245  E49C  00 00 00 00
1246  E4A0  00 00
1247  E4A2  AE E7                           .WORD   SYSVBL          ;VVBLKI
1248  E4A4  05 E9                           .WORD   XITVBL          ;VVBLKD
1249                                ;
1250                                        *=$900C
1251                                ;
1252  900C  A9 E6                           LDA     #PIRQH          ;SET UP RAM VECTORS FOR LINBUG VERSION
1253  900E  8D F9 FF                        STA     $FFF9
1254  9011  A9 F3                           LDA     #PIRQL
1255  9013  8D F8 FF                        STA     $FFF8
1256  9016  A9 E7                           LDA     #PNMIH
1257  9018  8D FB FF                        STA     $FFFB
1258  901B  A9 91                           LDA     #PNMIL
1259  901D  8D FA FF                        STA     $FFFA
1260  9020  60                              RTS
```

```
1261                            .PAGE
1262                    ;
1263                    ; IRQ HANDLER
1264                    ;
1265                    ; JUMP THRU IMMEDIATE IRQ VECTOR, WHICH ORDINARILY POINTS TO
1266                    ; SYSTEM IRQ; DETERMINE & CLEAR CAUSE, JUMP THRU SOFTWARE VECTOR.
1267                    ;
1268                            *=INTORG
1269  E6D5  A9 40       IHINIT: LDA     #$40            ;VBL ON BUF DLIST OFF***FOR NOW***
1270  E6D7  8D 0E D4            STA     NMIEN           ;ENABLE DISPLAY LIST, VERTICAL BLANK
1271  E6DA  A9 38               LDA     #$38            ;LOOK AT DATA DIRECTION REGISTERS IN PIA
1272  E6DC  8D 02 D3            STA     PACTL
1273  E6DF  8D 03 D3            STA     PBCTL
1274  E6E2  A9 00               LDA     #0              ;MAKE ALL INPUTS
1275  E6E4  8D 00 D3            STA     PORTA
1276  E6E7  8D 01 D3            STA     PORTB
1277  E6EA  A9 3C               LDA     #$3C            ;BACK TO PORTS
1278  E6EC  8D 02 D3            STA     PACTL
1279  E6EF  8D 03 D3            STA     PBCTL
1280  E6F2  60                  RTS
1281  E6F3  6C 16 02    PIRQ:   JMP     (VIMIRQ)
1282  E6F6  80          CMPTAB: .BYTE   $80             ;BREAK KEY
1283  E6F7  40                  .BYTE   $40             ;KEY STROKE
1284  E6F8  04                  .BYTE   $04             ;TIMER 4
1285  E6F9  02                  .BYTE   $02             ;TIMER 2
1286  E6FA  01                  .BYTE   $01             ;TIMER 1
1287  E6FB  08                  .BYTE   $08             ;SERIAL OUT COMPLETE
1288  E6FC  10                  .BYTE   $10             ;SERIAL OUT READY
1289  E6FD  20                  .BYTE   $20             ;SERIAL  IN READY
1290  E6FE
1291                    ; THIS IS A TABLE OF OFFSETS INTO PAGE 2.  THEY POINT TO
1292  E6FE  36          ADRTAB: .BYTE   BRKKY-INTABS
1293  E6FF  08                  .BYTE   VKEYBD-INTABS
1294  E700  14                  .BYTE   VTIMR4-INTABS
1295  E701  12                  .BYTE   VTIMR2-INTABS
1296  E702  10                  .BYTE   VTIMR1-INTABS
1297  E703  0E                  .BYTE   VSEROC-INTABS
1298  E704  0C                  .BYTE   VSEROR-INTABS
1299  E705  0A                  .BYTE   VSERIN-INTABS
1300  E706
1301  E706  48          SYIRQ:  PHA                     ;SAVE ACCUMULATOR
1302  E707  AD 0E D2            LDA     IRQST           ; CHECK FOR SERIAL IN
1303  E70A  29 20               AND     #$20
1304  E70C  D0 0D               BNE     SYIRQ2
1305  E70E  A9 DF               LDA     #$DF            ; MASK ALL OTHERS
1306  E710  8D 0E D2            STA     IRQEN
1307  E713  A5 10               LDA     POKMSK
1308  E715  8D 0E D2            STA     IRQEN
1309  E718  6C 0A 02            JMP     (VSERIN)
1310  E71B  8A          SYIRQ2: TXA                     ;PUT X INTO ACC
1311  E71C  48                  PHA                     ;SAVE X ONTO STACK
1312  E71D  A2 06               LDX     #$6             ;START WITH SIX OFFSET
1313  E71F  BD F6 E6    LOOPM:  LDA     CMPTAB,X        ;LOAD MASK
1314  E722  E0 05               CPX     #5              ;CHECK TO SEE IF COMPLETE IS SET
```

```
1315  E724  D0 04          BNE     LOOPM2
1316  E726  25 10          AND     POKMSK      ;IS THIS INTERUPT ENABLED?
1317  E728  F0 05          BEQ     LL
1318  E72A  2C 0E D2  LOOPM2: BIT   IRQST       ; IS IT THE INTERUPT?
1319  E72D  F0 06          BEQ     JMPP
1320  E72F  CA            LL:     DEX         ;NO DEC X AND TRY NEXT MASK
1321  E730  10 ED          BPL     LOOPM       ;IF NOT NEG GOTO LOOPM
1322  E732  4C 62 E7       JMP     SYIRQB      ;DONE BUT NO INTERUPT
1323  E735  49 FF          JMPP:   EOR     #$FF        ;COMPLEMENT MASK
1324  E737  8D 0E D2       STA     IRQEN       ;ENABLE ALL OTHERS
1325  E73A  A5 10          LDA     POKMSK      ; GET POKE MASK
1326  E73C  8D 0E D2       STA     IRQEN       ; ENABLE THOSE IN POKE MASK
1327  E73F  BD FE E6       LDA     ADRTAB,X
1328  E742  AA            TAX
1329  E743  BD 00 02       LDA     INTABS,X    ; GET ADDRESS LOW PART
1330  E746  8D 8C 02       STA     JVECK       ; PUT IN VECTOR
1331  E749  BD 01 02       LDA     INTABS+1,X  ; GET ADDRESS HIGH PART
1332  E74C  8D 8D 02       STA     JVECK+1     ; PUT IN VECTOR HIGH PART
1333  E74F  68            PLA                 ; PULL X REGISTER FROM STACK
1334  E750  AA            TAX                 ; PUT IT INTO X
1335  E751  6C 8C 02       JMP     (JVECK)     ; JUMP TO THE PROPER ROUTINE
1336  E754  A9 00          BRKKY2: LDA     #0          ; BREAK KEY ROUTINE
1337  E756  85 11          STA     BRKKEY      ; SET BREAK KEY FLAG
1338  E758  8D FF 02       STA     SSFLAG      ; START/STOP FLAG
1339  E75B  8D F0 02       STA     CRSINH      ; CURSOR INHIBIT
1340  E75E  85 4D          STA     ATRACT      ; TURN OFF ATRACT MODE
1341  E760  68            PLA
1342  E761  40            RTI                 ;EXIT FROM INT
1343  E762  68            SYIRQ8: PLA
1344  E763  AA            TAX
1345  E764  2C 02 D3       BIT     PACTL       ;PROCEED ***I GUESS***
1346  E767  10 06          BPL     SYIRQ9
1347  E769  AD 00 D3       LDA     PORTA       CLEAR INT STATUS BIT
1348  E76C  6C 02 02       JMP     (VPRCED)
1349  E76F  2C 03 D3       SYIRQ9: BIT     PBCTL       ;INTERRUPT ***I GUESS***
1350  E772  10 06          BPL     SYIRQA
1351  E774  AD 01 D3       LDA     PORTB       ;CLEAR INT STATUS
1352  E777  6C 04 02       JMP     (VINTER)
1353  E77A  68            SYIRQA: PLA
1354  E77B  8D 8C 02       STA     JVECK
1355  E77E  68            PLA
1356  E77F  48            PHA
1357  E780  29 10          AND     #$10        ;B BIT OF P REGISTER
1358  E782  F0 07          BEQ     SYRTI2
1359  E784  AD 8C 02       LDA     JVECK
1360  E787  48            PHA
1361  E788  6C 06 02       JMP     (VBREAK)
1362  E78B  AD 8C 02       SYRTI2: LDA     JVECK
1363  E78E  48            PHA
1364  E78F  68            SYIRQB: PLA
1365  E790  40            SYRTI:  RTI             ;UNIDENTIFIED INTERRUPT, JUST RETURN.
```

```
                                        .PAGE
                                ;
                                ; NMI HANDLER
                                ;
                                ; DETERMINE CAUSE AND JUMP THRU VECTOR
                                ;
E791  2C 0F D4                  PNMI:   BIT     NMIST
E794  10 03                             BPL     PNMI1           ;SEE IF DISPLAY LIST
E796  6C 00 02                          JMP     (VDSLST)
E799  48                        PNMI1:  PHA
E79A  AD 0F D4                          LDA     NMIST
E79D  29 20                             AND     #$20            ;SEE IF RESET
E79F  F0 03                             BEQ     *+5
E7A1  4C 74 E4                          JMP     WARMSV          ;GO THRU WARM START JUMP
E7A4  8A                                TXA                     ;SAVE REGISTERS
E7A5  48                                PHA
E7A6  98                                TYA
E7A7  48                                PHA
E7A8  8D 0F D4                          STA     NMIRES          ;RESET INTERRUPT STATUS
E7AB  6C 22 02                          JMP     (VVBLKI)        ;JUMP THRU VECTOR
```

```
1386                            .PAGE
1387                    ;
1388                    ; SYSTEM VBLANK ROUTINE
1389                    ;
1390                    ; INC FRAME COUNTER. PROCESS COUNTDOWN TIMERS. EXIT IF I WAS SET, CLEAR
1391                    ; SET DLISTL, DLISTH, DMACTL FROM RAM CELLS. DO SOFTWARE REPEAT.
1392                    ;
1393 E7AE E6 14    SYSVBL: INC     RTCLOK+2        ;INC FRAME COUNTER
1394 E7B0 D0 08            BNE     SYSVB1
1395 E7B2 E6 4D            INC     ATRACT          ;INCREMENT ATRACT (CAUSES ATRACT WHEN MINUS)
1396 E7B4 E6 13            INC     RTCLOK+1
1397 E7B6 D0 02            BNE     SYSVB1
1398 E7B8 E6 12            INC     RTCLOK
1399 E7BA A9 FE    SYSVB1: LDA     #$FE            ;(ATRACT) SET DARK MASK TO NORMAL
1400 E7BC A2 00            LDX     #0              ;SET COLRSH TO NORMAL
1401 E7BE A4 4D            LDY     ATRACT          ;TEST ATRACT FOR NEGATIVE
1402 E7C0 10 06            BPL     VBATRA          ;WHILE POSITIVE, DONT GO INTO ATRACT
1403 E7C2 85 4D            STA     ATRACT          ;IN ATRACT, SO STAY BY STA $FE
1404 E7C4 A6 13            LDX     RTCLOK+1        ;COLOR SHIFT FOLLOWS RTCLOK+1
1405 E7C6 A9 F6            LDA     #$F6            ;SET DARK MASK TO DARK
1406 E7C8 85 4E    VBATRA: STA     DRKMSK
1407 E7CA 86 4F            STX     COLRSH
1408 E7CC A2 00            LDX     #0              ;POINT TO TIMER1
1409 E7CE 20 D0 E8         JSR     DCTIMR          ;GO DECREMENT TIMER1
1410 E7D1 D0 03            BNE     SYSVB2          ;BRANCH IF STILL COUNTING
1411 E7D3 20 CA E8         JSR     JTIMR1          ;GO JUMP TO ROUTINE
1412 E7D6 A5 42    SYSVB2: LDA     CRITIC
1413 E7D8 D0 08            BNE     XXIT            ;GO IF CRITICAL SET
1414 E7DA BA               TSX                     ;SEE IF I WAS SET
1415 E7DB BD 04 01         LDA     $104,X          ;GET STACKED P
1416 E7DE 29 04            AND     #$04            ;I BIT
1417 E7E0 F0 03            BEQ     SYSVB3          ;BRANCH IF OK
1418 E7E2 4C 05 E9 XXIT:   JMP     XITVBL          ;I WAS SET, EXIT
1419 E7E5 AD 0D D4 SYSVB3: LDA     PENV
1420 E7E8 8D 35 02         STA     LPENV
1421 E7EB AD 0C D4         LDA     PENH
1422 E7EE 8D 34 02         STA     LPENH
1423 E7F1 AD 31 02         LDA     SDLSTH
1424 E7F4 8D 03 D4         STA     DLISTH
1425 E7F7 AD 30 02         LDA     SDLSTL
1426 E7FA 8D 02 D4         STA     DLISTL
1427 E7FD AD 2F 02         LDA     SDMCTL
1428 E800 8D 00 D4         STA     DMACTL
1429 E803 AD 6F 02         LDA     GPRIOR          ;GLOBAL PRIOR
1430 E806 8D 1B D0         STA     PRIOR
1431 E809 A2 08            LDX     #$08            ;TURN OFF KEYBOARD SPEAKER
1432 E80B 8E 1F D0         STX     CONSOL
1433 E80E 58       SCOLLP: CLI                     ;DISABLE INTERUPTS
1434 E80F BD C0 02         LDA     PCOLR0,X        ;LOAD COLOR REGISTERS FROM RAM
1435 E812 45 4F            EOR     COLRSH          ;DO COLOR SHIFT
1436 E814 25 4E            AND     DRKMSK          ;AND DARK ATRACT
1437 E816 9D 12 D0         STA     COLPM0,X
1438 E819 CA               DEX
1439 E81A 10 F2            BPL     SCOLLP
```

```
1440 E81C AD F4 02          LDA   CHBAS
1441 E81F 8D 09 D4          STA   CHBASE
1442 E822 AD F3 02          LDA   CHACT
1443 E825 8D 01 D4          STA   CHACTL
1444 E828 A2 02             LDX   #2          ;POINT TO TIMER 2
1445 E82A 20 D0 E8          JSR   DCTIMR
1446 E82D D0 03             BNE   SYSVB4      ;IF DIDNT GO ZERO
1447 E82F 20 CD E8          JSR   JTIMR2      ;GO JUMP TO TIMER2 ROUTINE
1448 E832 A2 02       SYSVB4: LDX   #2          ;RESTORE X
1449 E834 E8          SYSVBB: INX
1450 E835 E8                  INX
1451 E836 BD 18 02            LDA   CDTMV1,X
1452 E839 1D 19 02            ORA   CDTMV1+1,X
1453 E83C F0 06               BEQ   SYSVBA
1454 E83E 20 D0 E8            JSR   DCTIMR      ;DECREMENT AND SET FLAG IF NONZERO
1455 E841 9D 26 02            STA   CDTMF3-4,X
1456 E844 E0 08       SYSVBA: CPX   #8          ;SEE IF DONE ALL 3
1457 E846 D0 EC               BNE   SYSVBB      ;LOOP
1458                  ; CHECK DEBOUNCE COUNTER
1459 E848 AD 0F D2            LDA   SKSTAT
1460 E84B 29 04               AND   #$04        ;KEY DOWN BIT
1461 E84D F0 08               BEQ   SYVB6A      ;IF KEY DOWN
1462                  ; KEY UP SO COUNT IT
1463 E84F AD F1 02            LDA   KEYDEL      ;KEY DELAY COUNTER
1464 E852 F0 03               BEQ   SYVB6A      ;IF COUNTED DOWN ALREADY
1465 E854 CE F1 02            DEC   KEYDEL      ;COUNT IT
1466                  ; CHECK SOFTWARE REPEAT TIMER
1467 E857 AD 2B 02    SYVB6A: LDA   SRTIMR
1468 E85A F0 17               BEQ   SYSVB7      ;DOESN'T COUNT
1469 E85C AD 0F D2            LDA   SKSTAT
1470 E85F 29 04               AND   #$04        ;CHECK KEY DOWN BIT
1471 E861 D0 60               BNE   SYSVB6      ;BRANCH IF NO LONGER DOWN
1472 E863 CE 2B 02            DEC   SRTIMR      ;COUNT FRAME OF KEY DOWN
1473 E866 D0 0B               BNE   SYSVB7      ;BRANCH IF NOT RUN OUT
1474                  ; TIMER RAN OUT - RESET AND SIMULATE KEYBOARD IRQ
1475 E868 A9 06               LDA   #SRTIM2     ;TIMER VALUE
1476 E86A 8D 2B 02            STA   SRTIMR      ;SET TIMER
1477 E86D AD 09 D2            LDA   KBCODE      ;GET THE KEY
1478 E870 8D FC 02            STA   CH          ;PUT INTO CH
1479                  ; READ GAME CONTROLLERS
1480 E873 A0 01       SYSVB7: LDY   #1
1481 E875 A2 03               LDX   #3
1482 E877 B9 00 D3    STLOOP: LDA   PORTA,Y
1483 E87A 4A                  LSR   A
1484 E87B 4A                  LSR   A
1485 E87C 4A                  LSR   A
1486 E87D 4A                  LSR   A
1487 E87E 9D 78 02            STA   STICK0,X    ;STORE JOYSTICK
1488 E881 CA                  DEX
1489 E882 B9 00 D3            LDA   PORTA,Y
1490 E885 29 0F               AND   #$F
1491 E887 9D 78 02            STA   STICK0,X    ;STORE JOYSTICK
1492 E88A CA                  DEX
1493 E88B 88                  DEY
```

```
1494  E88C  10 E9              BPL     STLOOP
1495                    ;
1496  E88E  A2 03              LDX     #3
1497  E890  BD 10 D0    STRL:  LDA     TRIG0,X      ;MOVE JOYSTICK TRIGGERS
1498  E893  9D 84 02           STA     STRIG0,X
1499  E896  BD 00 D2           LDA     POT0,X       ;MOVE POT VALUES
1500  E899  9D 70 02           STA     PADDL0,X
1501  E89C  BD 04 D2           LDA     POT4,X
1502  E89F  9D 74 02           STA     PADDL4,X
1503  E8A2  CA                 DEX
1504  E8A3  10 EB              BPL     STRL
1505  E8A5  8D 0B D2           STA     POTGO        ;START POTS FOR NEXT TIME
1506                    ;
1507  E8A8  A2 06              LDX     #6
1508  E8AA  A0 03              LDY     #3
1509  E8AC  B9 78 02    PTRLP: LDA     STICK0,Y     ;TRANSFER BITS FROM JOYSTICKS
1510  E8AF  4A                 LSR     A            ;TO PADDLE TRIGGERS
1511  E8B0  4A                 LSR     A
1512  E8B1  4A                 LSR     A
1513  E8B2  9D 7D 02           STA     PTRIG1,X
1514  E8B5  A9 00              LDA     #0
1515  E8B7  2A                 ROL     A
1516  E8B8  9D 7C 02           STA     PTRIG0,X
1517  E8BB  CA                 DEX
1518  E8BC  CA                 DEX
1519  E8BD  88                 DEY
1520  E8BE  10 EC              BPL     PTRLP
1521                    ;
1522  E8C0  6C 24 02           JMP     (VVBLKD)     ;GO TO DEFERRED VBLANK ROUTINE
1523  00E8              SV7H   =       SYSVB7/256
1524  0073              SV7L   =       (-256)*SV7H+SYSVB7
1525  E8C3  A9 00       SYSVB6: LDA    #0
1526  E8C5  8D 2B 02           STA     SRTIMR       ;ZERO TIMER
1527  E8C8  F0 A9              BEQ     SYSVB7       ;UNCOND
1528  E8CA  6C 26 02    JTIMR1: JMP    (CDTMA1)
1529  E8CD  6C 28 02    JTIMR2: JMP    (CDTMA2)
1530                    ;
1531                    ; SUBROUTINE TO DECREMENT A COUNTDOWN TIMER
1532                    ; ENTRY X=OFFSET FROM TIMER 1
1533                    ; EXIT A,P=ZERO IF WENT ZERO, FF OTHERWISE
1534                    ;
1535  E8D0  BC 18 02    DCTIMR: LDY    CDTMV1,X     ;LO BYTE
1536  E8D3  D0 08              BNE     DCTIM1       ;NONZERO, GO DEC IT
1537  E8D5  BC 19 02           LDY     CDTMV1+1,X   ;SEE IF BOTH ZERO
1538  E8D8  F0 10              BEQ     DCTXF        ;YES, EXIT NONZERO
1539  E8DA  DE 19 02           DEC     CDTMV1+1,X   ;DEC HI BYTE
1540  E8DD  DE 18 02    DCTIM1: DEC    CDTMV1,X     ;DEC LO BYTE
1541  E8E0  D0 08              BNE     DCTXF
1542  E8E2  BC 19 02           LDY     CDTMV1+1,X
1543  E8E5  D0 03              BNE     DCTXF
1544  E8E7  A9 00              LDA     #0           ;WENT ZERO, RETURN ZERO
1545  E8E9  60                 RTS
1546  E8EA  A9 FF       DCTXF: LDA     #$FF         ;RETURN NONZERO
1547  E8EC  60                 RTS
```

```
1548                            .PAGE
1549                    ;
1550                    ; SUBROUTINE TO SET VERTICAL BLANK VECTORS AND TIMERS
1551                    ; ENTRY X=HI,Y=LO BYTE TO SET
1552                    ;       A= 1-5 TIMERS 1-5
1553                    ;          6 IMM VBLANK
1554                    ;          7 DEF VBLANK
1555                    ;
1556 E8ED 0A          SETVBL: ASL     A               ;MUL BY 2
1557 E8EE 8D 2D 02             STA     INTEMP
1558 E8F1 8A                   TXA
1559 E8F2 A2 05                LDX     #5
1560 E8F4 8D 0A D4             STA     WSYNC           ;WASTE 20 CPU CYCLES
1561 E8F7 CA           SETLOP: DEX                     ;TO ALOWD VBLANK TO HAPPEN
1562 E8F8 D0 FD                BNE     SETLOP          ;IF THIS IS LINE "7C"
1563 E8FA AE 2D 02             LDX     INTEMP
1564 E8FD 9D 17 02             STA     CDTMV1-1,X
1565 E900 98                   TYA
1566 E901 9D 16 02             STA     CDTMV1-2,X
1567 E904 60                   RTS
1568                    ;
1569                    ; EXIT FROM VERTICAL BLANK
1570                    ;
1571 E905 68           XITVBL: PLA                     ;UNSTACK Y
1572 E906 A8                   TAY
1573 E907 68                   PLA                     ;UNSTACK X
1574 E908 AA                   TAX
1575 E909 68                   PLA                     ;UNSTACK A
1576 E90A 40                   RTI                     ;AND GO BACK FROM WHENCE.
1577 00E6              PIRQH   =       PIRQ/256
1578 00F3              PIRQL   =       (-256)*PIRQH+PIRQ
1579 00E7              PNMIH   =       PNMI/256
1580 0091              PNMIL   =       (-256)*PNMIH+PNMI
1581                    ; SPARE BYTE OR MODULE TOO LONG FLAG
1582 E90B              CRNTP2  =*
1583                            *=$14
1584 0014 39           INTSPR: .BYTE   SIOORG-CRNTP2 ;^GINTHV IS TOO LONG
```

```
1585
1586                    .TITLE  'SIO ( SERIAL BUS INPUT/OUTPUT CONTROLLER )'
1587                    COLLEEN OPERATING SYSTEM
1588                    ;
1589                    ; SIO  ( SERIAL BUS INPUT/OUTPUT CONTROLLER )
1590                    ; WITH SOFTWARE BAUD RATE CORRECTION ON CASSETTE
1591                    ;
1592                    ;
1593                    ;       AL MILLER       3-APR-79
1594                    ;
1595                    ;
1596                    ; THIS MODULE HAS ONE ENTRY POINT.  IT IS CALLED BY THE DEVICE
1597                    ; HANDLERS.  IT INTERPRETS A PREVIOUSLY ESTABLISHED DEVICE CONTROL
1598                    ; BLOCK (STORED IN GLOBAL RAM) TO ISSUE COMMANDS
1599                    ; TO THE SERIAL BUS TO CONTROL TRANSMITTING AND RECEIVING DATA.
1600                    ;
1601                    ;
1602                    ;
1603                    ;
```

```
1604                            .PAGE
1605                    ; EQUATES
1606                    ;
1607                    ; DCD DEVICE BUS ID NUMBERS
1608  0030              FLOPPY  =       $30
1609                    ;PRINTR =       $40
1610                    ;CASSET =       $60             ;!!!!! *****
1611  0060              CASET   =       $60             ;!!!!! *****
1612                    ;
1613                    ;
1614                    ;BUS COMMANDS
1615                    ;
1616  0052              READ    =       'R
1617  0057              WRITE   =       'W
1618                    ;STATIS = 'S
1619                    ;FORMAT = '!
1620                    ;
1621                    ;
1622                    ; COMMAND AUX BYTES
1623                    ;
1624  0053              SIDWAY  =       'S              ;PRINT 16 CHARACTERS SIDEWAYS
1625  004E              NORMAL  =       'N              ;PRINT 40 CHARACTERS NORMALLY
1626  0044              DOUBLE  =       'D              ;PRINT 20 CHARACTERS DOUBLE WIDE
1627  0050              PLOT    =       'P              ;PLOT MODE
1628                    ;
1629                    ;
1630                    ; BUS RESPONSES
1631                    ;
1632  0041              ACK     =       'A              ;DEVICE ACKNOWLEDGES INFORMATION
1633  004E              NACK    =       'N              ;DEVICE DID NOT UNDERSTAND
1634  0043              COMPLT  =       'C              ;DEVICE SUCCESSFULLY COMPLETED OPERATION
1635  0045              ERROR   =       'E              ;DEVICE INCURRED AN ERROR IN AN ATTEMPTED OP
1636                    ;
1637                    ;
1638                    ; MISCELLANEOUS EQUATES
1639                    ;
1640  0028              B192LO  =       $28             ;19200 BAUD RATE POKEY COUNTER VALUES (LO BY
1641  0000              B192HI  =       $00             ;(HI BYTE)
1642  00CC              B600LO  =       $CC             ;600 BAUD (LO BYTE)
1643  0005              B600HI  =       $05             ;(HI BYTE)
1644  0005              HITONE  =       $05             ;FSK HI FREQ POKEY COUNTE VALUE (5326 HZ)
1645  0007              LOTONE  =       $07             ;FSK LO FREQ POKEY COUNTER VALUE (3995 HZ)
1646                    ;
1647                            .IF     PALFLG
1648                    WIRGLO  =       150             ;WRITE INTER RECORD GAP (IN 1/60 SEC)
1649                    RIRGLO  =       100             ;READ INTER RECORD GAP (IN 1/60 SEC)
1650                    WSIRG   =        13             ;SHORT WRITE INTER RECORD GAP
1651                    RSIRG   =         8             ;SHORT READ INTER RECORD GAP
1652                            .ENDIF
1653                            .IF     PALFLG-1
1654  00B4              WIRGLO  =       180             ;WRITE INTER RECORD GAP (IN 1/60 SEC)
1655  0078              RIRGLO  =       120             ;READ INTER RECORD GAP (IN 1/60 SEC)
1656  000F              WSIRG   =       15              ;SHORT WRITE INTER RECORD GAP
1657  000A              RSIRG   =       10              ;SHORT READ INTER RECORD GAP
```

```
1658                            .ENDIF
1659  0000                      WIRGHI  =       0
1660  0000                      RIRGHI  =       0
1661                            ;
1662  0034                      NCOMLO  =       $34             ;PIA COMMAND TO LOWER NOT COMMAND LINE
1663  003C                      NCOMHI  =       $3C             ;PIA COMMAND TO RAISE NOT COMMAND LINE
1664  0034                      MOTRQO  =       $34             ;PIA COMMAND TO TURN ON CASSETTE MOTOR
1665  003C                      MOTRST  =       $3C             ;PIA COMMAND TO TURN OFF MOTOR
1666                            ;
1667  0002                      TEMPHI  =       TEMP/256        ;ADDRESS OF TEMP CELL (HI BYTE)
1668  003E                      TEMPLO  =       (-256)*TEMPHI+TEMP ;(LO BYTE)
1669  0002                      CBUFHI  =       CDEVIC/256      ;ADDRESS OF COMMAND BUFFER (HI BYTE)
1670  003A                      CBUFLO  =       (-256)*CBUFHI+CDEVIC ;(LO BYTE)
1671                            ;
1672  000D                      CRETRI  =       13              ;NUMBER OF COMMAND FRAME RETRIES
1673  0001                      DRETRI  =       1               ;NUMBER OF DEVICE RETRIES
1674  0002                      CTIMLO  =       2               ;COMMAND FRAME ACK TIME OUT (LO BYTE)
1675  0000                      CTIMHI  =       0               ;COMMAND FRAME ACK TIME OUT (HI BYTE)
1676                            ;
1677                            ;
1678                            ;JTADRH  =       JTIMER/256     ;HI BYTE OF JUMP TIMER ROUTINE ADDR      "M
1679                            ;JTADRL  =       (-256)*JTADRH+JTIMER    ;"MOVED TO LINE 1428"
1680                            ;
```

```
1681                            .PAGE
1682                    ;       SIO
1683                    ;
1684                    ;
1685                            *=SIOV
1686  E459  4C 59 E9            JMP     SIO             ;SIO ENTRY POINT
1687                    ;
1688                            *=SIOINV
1689  E465  4C 44 E9            JMP     SIOINT          ;SIO INITIALIZATION ENTRY POINT
1690                    ;
1691                            *=SENDEV
1692  E468  4C F2 EB            JMP     SENDEN          ;SEND ENABLE ENTRY POINT
1693                    ;
1694                            *=VCTABL-INTABS+VSERIN
1695                    ;
1696  E48A  0F EB               .WORD   ISRSIR          ;VSERIN
1697  E48C  90 EA               .WORD   ISRODN          ;VSEROR
1698  E48E  CF EA               .WORD   ISRTD           ;VSEROC
1699                    ;
1700                    ;
1701                    ;
1702                            *=SIOORG
1703                    ;
1704                    ; SIO INITIALIZATION SUBROUTINE
1705                    ;
1706  E944  A9 3C       SIOINT: LDA     #MOTRST
1707  E946  8D 02 D3            STA     PACTL           ;TURN OFF MOTOR
1708                    ;
1709  E949  A9 3C               LDA     #NCOMHI
1710  E94B  8D 03 D3            STA     PBCTL           ;RAISE NOT COMMAND LINE
1711                    ;
1712                    ;
1713  E94E  A9 03               LDA     #3
1714  E950  8D 32 02            STA     SSKCTL          ;GET POKEY OUT OF INITIALIZE MODE
1715  E953  85 41               STA     SOUNDR          ;INIT POKE ADDRESS FOR QUIET I/O
1716  E955  8D 0F D2            STA     SKCTL
1717                    ;
1718                    ;
1719  E958  60                  RTS                     ;RETURN
1720                    ;
1721                    ;
1722                    ;
1723                    ;
1724                    ;
1725                    ;
1726  E959  BA          SIO:    TSX
1727  E95A  8E 18 03            STX     STACKP          ;SAVE STACK POINTER
1728  E95D  A9 01               LDA     #1
1729  E95F  85 42               STA     CRITIC
1730                    ;
1731  E961  AD 00 03            LDA     DDEVIC
1732  E964  C9 60               CMP     #CASET
1733  E966  D0 03               BNE     NOTCST          ;BRANCH IF NOT CASSETTE
1734  E968  4C 80 EB            JMP     CASENT          ;OTHERWISE JUMP TO CASSETTE ENTER
```

```
1735                            ;
1736                            ; ALL DEVICES EXCEPT CASSETTE ARE INTELLIGENT
1737                            ;
1738 E96B A9 00        NOTCST: LDA     #0
1739 E96D 8D 0F 03              STA     CASFLG          ;INIT CASSETTE FLAG TO NO CASSETTE
1740                            ;
1741 E970 A9 01                 LDA     #DRETRI         ;SET NUMBER OF DEVICE RETRIES
1742 E972 85 37                 STA     DRETRY
1743 E974 A9 0D         COMMND: LDA     #CRETRI         ;SET NUMBER OF COMMAND FRAME RETRIES
1744 E976 85 36                 STA     CRETRY
1745                            ;
1746                            ; SEND A COMMAND FRAME
1747                            ;
1748 E978 A9 28         COMFRM: LDA     #B192LO         ;SET BAUD RATE TO 19200
1749 E97A 8D 04 D2              STA     AUDF3
1750 E97D A9 00                 LDA     #B192HI
1751 E97F 8D 06 D2              STA     AUDF4
1752                            ;
1753 E982 18                    CLC                     ;SET UP COMMAND BUFFER
1754 E983 AD 00 03              LDA     DDEVIC
1755 E986 6D 01 03              ADC     DUNIT
1756 E989 69 FF                 ADC     #$FF            ;SUBTRACT 1
1757 E98B 8D 3A 02              STA     CDEVIC          ;SET BUS ID NUMBER
1758                            ;
1759 E98E AD 02 03              LDA     DCOMND
1760 E991 8D 3B 02              STA     CCOMND          ;SET BUS COMMAND
1761                            ;
1762 E994 AD 0A 03              LDA     DAUX1           ;STORE COMMAND FRAME AUX BYTES 1 AND 2
1763 E997 8D 3C 02              STA     CAUX1
1764 E99A AD 0B 03              LDA     DAUX2
1765 E99D 8D 3D 02              STA     CAUX2           ;DONE SETTING UP COMMAND BUFFER
1766                            ;
1767 E9A0 18                    CLC                     ;SET BUFFER POINTER TO COMMAND FRAME BUFFER
1768 E9A1 A9 3A                 LDA     #CBUFLO
1769 E9A3 85 32                 STA     BUFRLO          ;AND BUFFER END ADDRESS
1770 E9A5 69 04                 ADC     #4
1771 E9A7 85 34                 STA     BFENLO
1772 E9A9 A9 02                 LDA     #CBUFHI
1773 E9AB 85 33                 STA     BUFRHI
1774 E9AD 85 35                 STA     BFENHI          ;DONE SETTING UP BUFFER POINTER
1775                            ;
1776 E9AF A9 34                 LDA     #NCOMLO         ;LOWER NOT COMMAND LINE
1777 E9B1 8D 03 D3              STA     PBCTL
1778                            ;
1779 E9B4 20 8A EC              JSR     SENDIN          ;SEND THE COMMAND FRAME TO A SMART DEVICE
1780                            ;
1781 E9B7 AD 3F 02              LDA     ERRFLG
1782 E9BA D0 03                 BNE     BADCOM          ;BRANCH IF AN ERROR RECEIVED
1783                            ;
1784 E9BC 98                    TYA
1785 E9BD D0 07                 BNE     ACKREC          ;BRANCH IF ACK RECEIVED
1786                            ;
1787                            ;
1788 E9BF C6 36         BADCOM: DEC     CRETRY          ;A NACK OR TIME OUT OCCURED
```

```
1789  E9C1  10 B5          BPL     COMFRM      ;SO BRANCH IF ANY RETRIES LEFT
1790                       ;
1791  E9C3  4C 06 EA       JMP     DERR1       ;OTHERWISE, JUMP TO RETURN SECTION
1792                       ;
1793                       ;
1794  E9C6  AD 03 03       ACKREC: LDA     DSTATS      ;ACK WAS RECEIVED
1795  E9C9  10 0C          BPL     WATCOM      ;BRANCH TO WAIT FOR COMPLETE ,
1796                       ; IF THERE IS NO DATA TO BE SENT
1797                       ;
1798                       ;
1799                       ;
1800                       ; SEND A DATA FRAME TO PERIPHERAL
1801                       ;
1802  E9CB  A9 0D          LDA     #CRETRI     ;SET NUMBER OF RETRIES
1803  E9CD  85 36          STA     CRETRY
1804                       ;
1805  E9CF  20 6A EB       JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
1806                       ;
1807  E9D2  20 8A EC       JSR     SENDIN      ;GO SEND THE DATA FRAME TO A SMART DEVICE
1808                       ;
1809  E9D5  F0 E8          BEQ     BADCOM      ;BRANCH IF BAD
1810                       ;
1811                       ;
1812                       ;
1813                       ; WAIT FOR COMPLETE SIGNAL FROM PERIPHERAL
1814                       ;
1815  E9D7  20 75 EC       WATCOM: JSR     STTMOT      ;SET DDEVICE TIME OUT VALUES IN Y,X
1816                       ;
1817  E9DA  A9 00          LDA     #$00
1818  E9DC  8D 3F 02       STA     ERRFLG      ;CLEAR ERROR FLAG
1819                       ;
1820  E9DF  20 9B EC       JSR     WAITER      ;SET UP TIMER AND WAIT
1821  E9E2  F0 12          BEQ     DERR        ;BRANCH IF TIME OUT
1822                       ;
1823                       ;
1824                       ; DEVICE DID NOT TIME OUT
1825                       ;
1826  E9E4  2C 03 03       BIT     DSTATS
1827  E9E7  70 07          BVS     MODATA      ;BRANCH IF MORE DATA FOLLOWS
1828                       ;
1829  E9E9  AD 3F 02       LDA     ERRFLG
1830  E9EC  D0 18          BNE     DERR1       ;BRANCH IF AN ERROR OCCURRED
1831  E9EE  F0 1D          BEQ     RETURN      ;OTHERWISE RETURN
1832                       ;
1833                       ;
1834                       ;
1835                       ;
1836                       ; RECEIVE A DATA FRAME FROM PERIPHERAL
1837                       ;
1838  E9F0  20 6A EB       MODATA: JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
1839                       ;
1840  E9F3  20 E0 EA       JSR     RECEIV      ;GO RECEIVE A DATA FRAME
1841                       ;
1842  E9F6  AD 3F 02       DERR:   LDA     ERRFLG
```

```
1843  E9F9  F0 05          BEQ     NOTERR      ;BRANCH IF NO ERROR PRECEEDED DATA
1844                       ;
1845  E9FB  AD 19 03       LDA     TSTAT       ;GET TEMP STATUS
1846  E9FE  85 30          STA     STATUS      ;STORE IN REAL STATUS
1847                       ;
1848                       ;
1849  EA00  A5 30          NOTERR: LDA     STATUS
1850  EA02  C9 01                  CMP     #SUCCES
1851  EA04  F0 07                  BEQ     RETURN      ;BRANCH IF COMPLETELY SUCCESSFUL
1852                       ;
1853  EA06  C6 37          DERR1:  DEC     DRETRY
1854  EA08  30 03                  BMI     RETURN      ;BRANCH IF OUT OF DEVICE RETRIES
1855                       ;
1856  EA0A  4C 74 E9               JMP     COMMND      ;OTHERWISE, ONE MORE TIME
1857                       ;
1858                       ;
1859                       ;
1860                       ;
1861  EA0D  20 5F EC       RETURN: JSR     SENDDS      ;DISABLE POKEY INTERRUPTS
1862  EA10  A9 00                  LDA     #0
1863  EA12  85 42                  STA     CRITIC
1864  EA14  A4 30                  LDY     STATUS      ;RETURN STATUS IN Y
1865  EA16  8C 03 03               STY     DSTATS      ;AND THE DCB STATUS WORD
1866  EA19  60                     RTS     RETURN
1867                       ;
1868                       ;
1869                       ;
1870                       ;
1871                       ; WAIT SUBROUTINE
1872                       ;
1873                       ; WAITS FOR COMPLETE OR ACK
1874                       ; RETURNS Y=$FF IF SUCCESSFUL, Y=$00 IF NOT
1875                       ;
1876  EA1A  A9 00          WAIT:   LDA     #$00
1877  EA1C  8D 3F 02               STA     ERRFLG      ;CLEAR ERROR FLAG
1878                       ;
1879  EA1F  18                     CLC                 ;LOAD BUFFER POINTER WITH ADDRESS
1880  EA20  A9 3E                  LDA     #TEMPLO     ;OF TEMPORARY RAM CELL
1881  EA22  85 32                  STA     BUFRLO
1882  EA24  69 01                  ADC     #1
1883  EA26  85 34                  STA     BFENLO      ;ALSO SET BUFFER END +1 ADDRESS
1884  EA28  A9 02                  LDA     #TEMPHI
1885  EA2A  85 33                  STA     BUFRHI
1886  EA2C  85 35                  STA     BFENHI      ;DONE LOADING POINTER
1887                       ;
1888  EA2E  A9 FF                  LDA     #$FF
1889  EA30  85 3C                  STA     NOCKSM      ;SET NO CHECKSUM FOLLOWS DATA FLAG
1890                       ;
1891  EA32  20 E0 EA               JSR     RECEIV      ;GO RECEIVE A BYTE
1892                       ;
1893  EA35  A0 FF                  LDY     #$FF        ;ASSUME SUCCESS
1894  EA37  A5 30                  LDA     STATUS
1895  EA39  C9 01                  CMP     #SUCCES
1896  EA3B  D0 19                  BNE     NWOK        ;BRANCH IF IT DID NOT WORK OK
```

```
1897                                ;
1898                                ;
1899                                ;
1900                                ;
1901 EA3D AD 3E 02          WOK:    LDA     TEMP            ;MAKE SURE THE BYTE SUCCESSFULLY RECEIVED
1902 EA40 C9 41                     CMP     #ACK            ;WAS ACTUALLY AN ACK OR COMPLETE
1903 EA42 F0 21                     BEQ     GOOD
1904 EA44 C9 43                     CMP     #COMPLT
1905 EA46 F0 1D                     BEQ     GOOD
1906                                ;
1907 EA48 C9 45                     CMP     #ERROR
1908 EA4A D0 06                     BNE     NOTDER          ;BRANCH IF DEVICE DID NOT SEND BACK
1909                                ; A DEVICE ERROR CODE
1910 EA4C A9 90                     LDA     #DERROR
1911 EA4E 85 30                     STA     STATUS          ;SET DEVICE ERROR STATUS
1912 EA50 D0 04                     BNE     NWOK
1913                                ;
1914 EA52 A9 8B            NOTDER: LDA     #DNACK          ;OTHERWISE SET NACK STATUS
1915 EA54 85 30                     STA     STATUS
1916                                ;
1917 EA56 A5 30             NWOK:   LDA     STATUS
1918 EA58 C9 8A                     CMP     #TIMOUT
1919 EA5A F0 07                     BEQ     BAD             ;BRANCH IF TIME OUT
1920                                ;
1921 EA5C A9 FF                     LDA     #$FF
1922 EA5E 8D 3F 02                  STA     ERRFLG          ;SET SOME ERROR FLAG
1923 EA61 D0 02                     BNE     GOOD            ;RETURN WITH OUT SETTING Y = 0
1924                                ;
1925 EA63 A0 00             BAD:    LDY     #0
1926                                ;
1927 EA65 A5 30             GOOD:   LDA     STATUS
1928 EA67 8D 19 03                  STA     TSTAT
1929 EA6A 60                        RTS                     ;RETURN
1930                                )
1931                                )
1932                                )
1933                                )
1934                                )
1935                                ) SEND SUBROUTINE
1936                                )
1937                                ) SENDS A BUFFER OF BYTES OUT OVER THE SERIAL BUS
1938                                )
1939                                )
1940 EA6B A9 01             SEND:   LDA     #SUCCES         ;ASSUME SUCCESS
1941 EA6D 85 30                     STA     STATUS
1942                                ;
1943 EA6F 20 F2 EB                  JSR     SENDEN          ;ENABLE SENDING
1944                                ;
1945 EA72 A0 00                     LDY     #0
1946 EA74 84 31                     STY     CHKSUM          ;CLEAR CHECK SUM
1947 EA76 84 3B                     STY     CHKSNT          ;CHECKSUM SENT FLAG
1948 EA78 84 3A                     STY     XMTDON          ;TRANSMISSION DONE FLAG
1949                                )
1950                                )
```

```
1951  EA7A  B1 32          LDA     (BUFRLO),Y  ;PUT FIRST BYTE FROM BUFFER
1952  EA7C  8D 0D D2       STA     SEROUT      ;INTO THE SERIAL OUTPUT REGISTER
1953                       ;
1954                       ;
1955  EA7F  85 31          STA     CHKSUM      ;PUT IT IN CHECKSUM
1956                       ;
1957  EA81  A5 11          NOTDON: LDA     BRKKEY
1958  EA83  D0 03          BNE     NTBRKO
1959  EA85  4C A0 ED       JMP     BROKE       ;JUMP IF BREAK KEY PRESSED
1960                       ;
1961  EA88  A5 3A          NTBRKO: LDA     XMTDON      ;LOOP UNTIL TRANSMISSION IS DONE
1962  EA8A  F0 F5          BEQ     NOTDON
1963                       ;
1964  EA8C  20 5F EC       JSR     SENDDS      ;DISABLE SENDING
1965                       ;
1966  EA8F  60             RTS                 ;RETURN
1967                       ;
1968                       ;
1969                       ;
1970                       ;
1971                       ;
1972                       ;
1973                       ; OUTPUT DATA NEEDED INTERRUPT SERVICE ROUTINE
1974                       ;
1975  EA90  98             ISRODN: TYA
1976  EA91  48             PHA                 ;SAVE Y REG ON STACK
1977                       ;
1978  EA92  E6 32          INC     BUFRLO      ;INCREMENT BUFFER POINTER
1979  EA94  D0 02          BNE     NOWRPO
1980  EA96  E6 33          INC     BUFRHI
1981                       ;
1982  EA98  A5 32          NOWRPO: LDA     BUFRLO      ;CHECK IF PAST END OF BUFFER
1983  EA9A  C5 34          CMP     BFENLO
1984  EA9C  A5 33          LDA     BUFRHI      ;HIGH PART
1985  EA9E  E5 35          SBC     BFENHI
1986  EAA0  90 1C          BCC     NOTEND      ;BRANCH IF NOT PAST END OF BUFFER
1987                       ;
1988  EAA2  A5 3B          LDA     CHKSNT
1989  EAA4  D0 0B          BNE     RELONE      ;BRANCH IF CHECKSUM ALREADY SENT
1990                       ;
1991  EAA6  A5 31          LDA     CHKSUM
1992  EAA8  8D 0D D2       STA     SEROUT      ;SEND CHECK SUM
1993  EAAB  A9 FF          LDA     #$FF
1994  EAAD  85 3B          STA     CHKSNT      ;SET CHECKSUM SENT FLAG
1995  EAAF  D0 09          BNE     CHKDON
1996                       ;
1997  EAB1  A5 10          RELONE: LDA     POKMSK      ;ENABLE TRANSMIT DONE INTERRUPT
1998  EAB3  09 08          ORA     #$08
1999  EAB5  85 10          STA     POKMSK
2000  EAB7  8D 0E D2       STA     IRQEN
2001                       ;
2002  EABA  68             CHKDON: PLA
2003  EABB  A8             TAY                 ;RESTORE Y REG
2004  EABC  68             PLA                 ;RETURN FROM INTERRUPT
```

```
2005  EABD  40                RTI
2006                    ;
2007                    ;
2008  EABE  A0 00       NOTEND: LDY     #0
2009  EAC0  B1 32               LDA     (BUFRLO),Y  ;PUT NEXT BYTE FROM BUFFER
2010  EAC2  8D 0D D2            STA     SEROUT      ;INTO THE SERIAL OUTPUT REGISTER
2011                    ;
2012  EAC5  18                  CLC                 ;ADD IT TO CHECKSUM
2013  EAC6  65 31               ADC     CHKSUM
2014  EAC8  69 00               ADC     #0
2015  EACA  85 31               STA     CHKSUM
2016                    ;
2017  EACC  4C BA EA            JMP     CHKDON      ;GO RETURN
2018                    ;
2019                    ;
2020                    ;
2021                    ;
2022                    ;
2023                    ;
2024                    ; TRANSMIT DONE INTERRUPT SERVICE ROUTINE
2025                    ;
2026  EACF  A5 3B       ISRTD:  LDA     CHKSNT
2027  EAD1  F0 0B               BEQ     FOOEY       ;BRANCH IF CHECKSUM NOT YET SENT
2028                    ;
2029  EAD3  85 3A               STA     XMTDON      ;OTHERWISE SET TRANSMISSION DONE FLAG
2030                    ;
2031  EAD5  A5 10               LDA     POKMSK      ;DISABLE TRANSMIT DONE INTERRUPT
2032  EAD7  29 F7               AND     #$F7
2033  EAD9  85 10               STA     POKMSK
2034  EADB  8D 0E D2            STA     IRQEN
2035                    ;
2036  EADE  68          FOOEY:  PLA                 ;RETURN FROM INTERRUPT
2037  EADF  40                  RTI
2038                    ;
2039                    ;
2040                    ;
2041                    ;
2042                    ;
2043                    ;
2044                    ;
2045                    ;
2046                    ; RECEIVE SUBROUTINE
2047                    ;
2048  EAE0  A9 00       RECEIV: LDA     #0
2049                    ;
2050  EAE2  AC 0F 03            LDY     CASFLG
2051  EAE5  D0 02               BNE     NOCLR       ;BRANCH IF CASSETTE
2052                    ;
2053  EAE7  85 31               STA     CHKSUM      ;CLEAR CHKSUM
2054  EAE9  85 38       NOCLR:  STA     BUFRFL      ;BUFFER FULL FLAG
2055  EAEB  85 39               STA     RECVDN      ;RECEIVE DONE FLAG
2056                    ;
2057                    ;
2058                    ;
```

```
2059  EAED  A9 01          LDA     #SUCCES
2060  EAEF  85 30          STA     STATUS      ;SET GOOD STATUS FOR DEFAULT CASE.
2061  EAF1  20 1B EC       JSR     RECVEN      ;DO RECEIVE ENABLE
2062  EAF4  A9 3C          LDA     #NCOMHI     ;COMMAND FRAME HI COMMAND
2063  EAF6  8D 03 D3       STA     PBCTL       ;STORE IN PIA
2064  EAF9  A5 11       CHKTIM: LDA     BRKKEY
2065  EAFB  D0 03          BNE     NTBRK1
2066  EAFD  4C A0 ED       JMP     BROKE       ;JUMP IF BREAK KEY PRESSED
2067                     ;
2068  EB00  AD 17 03    NTBRK1: LDA     TIMFLG      ;NO,
2069  EB03  F0 05          BEQ     TOUT        ;IF TIMEOUT, GO SET ERROR STATUS
2070  EB05  A5 39          LDA     RECVDN
2071  EB07  F0 F0          BEQ     CHKTIM      ;DONE ?
2072  EB09  60          GOBACK: RTS
2073  EB0A  A9 8A       TOUT:   LDA     #TIMOUT     ;YES,
2074  EB0C  85 30          STA     STATUS      ;SET TIMEOUT STATUS
2075                     ;
2076                     ;
2077                     ;
2078                     ;
2079                     ;
2080                     ;
2081  EB0E  60           RRETRN: RTS             ;RETURN
2082                     ;
2083                     ;
2084                     ;
2085                     ;
2086                     ;
2087                     ;
2088                     ;
2089                     ; SERIAL INPUT READY INTERRUPT SERVICE ROUTINE
2090                     ;
2091  EB0F  98           ISRSIR: TYA
2092  EB10  48                   PHA             ;SAVE Y REG ON STACK
2093                     ;
2094                     ;
2095                     ;
2096  EB11  AD 0F D2             LDA     SKSTAT
2097  EB14  8D 0A D2             STA     SKRES       ;RESET STATUS REGISTER
2098                     ; ********   THIS MAY NOT BE THE PLACE TO DO IT   *********
2099                     ;
2100  EB17  30 04                BMI     NTFRAM      ;BRANCH IF NO FRAMING ERROR
2101                     ;
2102  EB19  A0 8C                LDY     #FRMERR
2103  EB1B  84 30                STY     STATUS      ;SET FRAME ERRORR STATUS
2104                     ;
2105  EB1D  29 20        NTFRAM: AND     #$20
2106  EB1F  D0 04                BNE     NTOVRN      ;BRANCH IF NO OVERRUN ERROR
2107                     ;
2108  EB21  A0 8E                LDY     #OVRRUN
2109  EB23  84 30                STY     STATUS      ;SET OVERRUN ERROR STATUS
2110                     ;
2111  EB25  A5 38        NTOVRN: LDA     BUFRFL
2112  EB27  F0 13                BEQ     NOTYET      ;BRANCH IF BUFFER WAS NOT YET FILLED
```

```
2113                            ;
2114  EB29  AD 0D D2           LDA     SERIN       ;THIS INPUT BYTE IS THE CHECKSUM
2115  EB2C  C5 31               CMP     CHKSUM
2116  EB2E  F0 04               BEQ     SRETRN      ;BRANCH IF CHECKSUMS MATCH
2117                            ;
2118  EB30  A0 8F               LDY     #CHKERR
2119  EB32  84 30               STY     STATUS      ;SET CHECKSUM ERROR STATUS
2120                            ;
2121  EB34  A9 FF     SRETRN:   LDA     #$FF        ;SET RECEIVE DONE FLAG
2122  EB36  85 39               STA     RECVDN
2123                            ;
2124  EB38  68        SUSUAL:   PLA
2125  EB39  A8                  TAY                 ;RESTORE Y REG
2126  EB3A  68                  PLA                 ;RETURN FROM INTERRUPT
2127  EB3B  40                  RTI
2128                            ;
2129                            ;
2130                            ;
2131  EB3C  AD 0D D2   NOTYET:  LDA     SERIN
2132  EB3F  A0 00               LDY     #0
2133  EB41  91 32               STA     (BUFRLO),Y  ;STORE INPUT REGISTER INTO BUFFER
2134                            ;
2135  EB43  18                  CLC                 ;ADD IT TO CHECKSUM
2136  EB44  65 31               ADC     CHKSUM
2137  EB46  69 00               ADC     #0
2138  EB48  85 31               STA     CHKSUM
2139                            ;
2140  EB4A  E6 32               INC     BUFRLO      ;INCREMENT BUFFER POINTER
2141  EB4C  D0 02               BNE     NTWRP1
2142  EB4E  E6 33               INC     BUFRHI
2143                            ;
2144  EB50  A5 32     NTWRP1:   LDA     BUFRLO
2145  EB52  C5 34               CMP     BFENLO
2146  EB54  A5 33               LDA     BUFRHI
2147  EB56  E5 35               SBC     BFENHI
2148  EB58  90 DE               BCC     SUSUAL      ;BRANCH IF NEW BUFFER ADDRESS IS IN BUFFER L
2149                            ;
2150  EB5A  A5 3C               LDA     NOCKSM
2151  EB5C  F0 06               BEQ     GOON        ;BRANCH IF A CHECKSUM WILL FOLLOW DATA
2152                            ;
2153  EB5E  A9 00               LDA     #0
2154  EB60  85 3C               STA     NOCKSM      ;CLEAR NO CHECKSUM FLAG
2155                            ;
2156  EB62  F0 D0               BEQ     SRETRN      ;GO RETURN AND SET RECEIVE DONE FLAG
2157                            ;
2158                            ;
2159  EB64  A9 FF     GOON:     LDA     #$FF
2160  EB66  85 38               STA     BUFRFL      ;SET BUFFER FULL FLAG
2161                            ;
2162  EB68  D0 CE               BNE     SUSUAL      ;GO RETURN
2163                            ;
2164                            ;
2165                            ;
2166                            ;
```

```
2167                                ;
2168                                ;
2169                                ;
2170                                ;
2171                                ; LOAD BUFFER POINTER SUBROUTINE
2172                                ;
2173                                ; LOAD BUFFER POINTER WITH DCB BUFFER INFORMATION
2174                                ;
2175  EB6A  18                     LDPNTR: CLC
2176  EB6B  AD 04 03                       LDA     DBUFLO
2177  EB6E  85 32                          STA     BUFRLO
2178  EB70  6D 08 03                       ADC     DBYTLO
2179  EB73  85 34                          STA     BFENLO      ;ALSO SET BUFFER END + 1 ADDRESS
2180                                ;
2181  EB75  AD 05 03                       LDA     DBUFHI
2182  EB78  85 33                          STA     BUFRHI
2183  EB7A  6D 09 03                       ADC     DBYTHI
2184  EB7D  85 35                          STA     BFENHI
2185                                ;
2186  EB7F  60                             RTS                 ;RETURN
2187                                ;
2188                                ;
2189                                ;
2190                                ;
2191                                ;
2192                                ;
2193                                ;
2194                                ;
2195                                ; CASSETTE HANDLING CODE
2196                                ;
2197  EB80  AD 03 03               CASENT: LDA     DSTATS
2198  EB83  10 2E                          BPL     CASRED      ;BRANCH IF INPUT FROM CASSETTE
2199                                ;
2200                                ; WRITE A RECORD
2201                                ;
2202  EB85  A9 CC                          LDA     #B600LO     ;SET BAUD RATE TO 600
2203  EB87  8D 04 D2                       STA     AUDF3
2204  EB8A  A9 05                          LDA     #B600HI
2205  EB8C  8D 06 D2                       STA     AUDF4
2206                                ;
2207  EB8F  20 F2 EB                       JSR     SENDEN      ;TURN ON POKEY MARK TONE
2208                                ;
2209  EB92  A0 0F                          LDY     #WSIRG      ;LOAD SHORT WRITE INTER RECORD GAP TIME
2210  EB94  AD 0B 03                       LDA     DAUX2
2211  EB97  30 02                          BMI     SRTIRO      ;BRANCH IF SHORT GAP IS DESIRED
2212                                ;
2213  EB99  A0 B4                          LDY     #WIRGLO     ;SET WRITE IRG TIME
2214  EB9B  A2 00                  SRTIRO: LDX     #WIRGHI
2215  EB9D  20 B9 ED                       JSR     SETVBX
2216                                ;
2217  EBA0  A9 34                          LDA     #MOTRGO
2218  EBA2  8D 02 D3                       STA     PACTL       ;TURN ON MOTOR
2219                                ;
2220  EBA5  AD 17 03               TIMIT:  LDA     TIMFLG      ;LOOP UNTIL DONE
```

```
2221  EBA8  D0 FB              BNE     TIMIT
2222                          ;
2223  EBAA  20 6A EB          JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
2224                          ;
2225  EBAD  20 6B EA          JSR     SEND        ;SEND A BUFFER
2226                          ;
2227  EBB0  4C DF EB          JMP     CRETRN      ;GO RETURN
2228                        ;
2229                        ;
2230                        ;
2231                        ; RECEIVE A RECORD
2232                        ;
2233  EBB3  A9 FF           CASRED: LDA     #$FF
2234  EBB5  8D 0F 03          STA     CASFLG      ;SET SET CASSETTE FLAG
2235                        ;
2236  EBB8  A0 0A             LDY     #RSIRG      ;LOAD SHORT READ INTER RECORD GAP TIME
2237  EBBA  AD 0B 03          LDA     DAUX2
2238  EBBD  30 02             BMI     SRTIR1      ;BRANCH IF SHORT GAP IS DESIRED
2239                        ;
2240  EBBF  A0 78             LDY     #RIRGLO     ;SET TIME OUT FOR READ IRG
2241  EBC1  A2 00           SRTIR1: LDX     #RIRGHI
2242  EBC3  20 B9 ED          JSR     SETVBX
2243                        ;
2244  EBC6  A9 34             LDA     #MOTRGO
2245  EBC8  8D 02 D3          STA     PACTL       ;TURN ON MOTOR
2246                        ;
2247  EBCB  AD 17 03        TIMIT1: LDA     TIMFLG      ;LOOP UNTIL DONE
2248  EBCE  D0 FB             BNE     TIMIT1
2249                        ;
2250  EBD0  20 6A EB          JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
2251                        ;
2252  EBD3  20 75 EC          JSR     STTMOT      ;SET DEVICE TIME OUT IN Y,X
2253  EBD6  20 B9 ED          JSR     SETVBX
2254                        ;
2255  EBD9  20 10 ED          JSR     BEGIN       ;SET INITIAL BAUD RATE
2256                        ;
2257  EBDC  20 E0 EA          JSR     RECEIV      ;GO RECEIVE A BLOCK
2258                        ;
2259  EBDF  AD 0B 03        CRETRN: LDA     DAUX2
2260  EBE2  30 05             BMI     SRTIR2      ;BRANCH IF DOING SHORT INTER RECORD GAPS
2261                        ; DON'T TURN OFF CASSETTE MOTOR
2262  EBE4  A9 3C             LDA     #MOTRST
2263  EBE6  8D 02 D3          STA     PACTL       ;TURN OFF MOTOR
2264                        ;
2265  EBE9  4C 0D EA        SRTIR2: JMP     RETURN      ;GO RETURN
2266                        ;
2267                        ;
2268                        ;
2269                        ;
2270                        ;
2271  EBEC  A9 00           JTIMER: LDA     #$00
2272  00EB                  JTADRH  =       JTIMER/256  ;HI BYTE OF JUMP TIMER ROUTINE ADDR
2273  00EC                  JTADRL  =       (-256)*JTADRH+JTIMER
2274  EBEE  8D 17 03          STA     TIMFLG      ;SET TIME OUT FLAG
```

```
2275  EBF1  60                RTS
2276
2277
2278
2279
2280
2281
2282                          ; SEND ENABLE SUBROUTINE
2283                          ;
2284  EBF2  A9 07             SENDEN: LDA     #$07        ;MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
2285  EBF4  2D 32 02                  AND     SSKCTL
2286  EBF7  09 20                     ORA     #$20        ;SET TRANSMIT MODE
2287                          ;
2288  EBF9  AC 00 03                  LDY     DDEVIC
2289  EBFC  C0 60                     CPY     #CASET
2290  EBFE  D0 0C                     BNE     NOTCAS      ;BRANCH IF NOT CASSETTE
2291                          ;
2292  EC00  09 08                     ORA     #$08        ;SET THE FSK OUTPUT BIT
2293                          ;
2294  EC02  A0 07                     LDY     #LOTONE     ;SET FSK TONE FREQUENCIES
2295  EC04  8C 02 D2                  STY     AUDF2
2296  EC07  A0 05                     LDY     #HITONE
2297  EC09  8C 00 D2                  STY     AUDF1
2298                          ;
2299  EC0C  8D 32 02          NOTCAS: STA     SSKCTL      ;STORE NEW VALUE TO SYSTEM MASK
2300  EC0F  8D 0F D2                  STA     SKCTL       ;STORE TO ACTUAL REGISTER
2301                          ;
2302  EC12  A9 C7                     LDA     #$C7        ;MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
2303  EC14  25 10                     AND     POKMSK
2304  EC16  09 10                     ORA     #$10        ;ENABLE OUTPUT DATA NEEDED INTERRUPT
2305
2306
2307  EC18  4C 31 EC                  JMP     CONTIN      ;GO CONTINUE IN RECEIVE ENABLE SUBROUTINE
2308                          ;
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318                          ; RECEIVE ENABLE SUBROUTINE
2319                          ;
2320  EC1B  A9 07             RECVEN: LDA     #$07        ;MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
2321  EC1D  2D 32 02                  AND     SSKCTL
2322  EC20  09 10                     ORA     #$10        ;SET RECEIVE MODE ASYNCH.
2323  EC22  8D 32 02                  STA     SSKCTL      ;STORE NEW VALUE TO SYSTEM MASK
2324  EC25  8D 0F D2                  STA     SKCTL       ;STORE TO ACTUAL REGISTER
2325                          ;
2326  EC28  8D 0A D2                  STA     SKRES       ;RESET SERIAL PORT/KEYBOARD STATUS REGISTER
2327                          ;
2328  EC2B  A9 C7                     LDA     #$C7        ;MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
```

```
2329  EC2D  25 10          AND     POKMSK
2330  EC2F  09 20          ORA     #$20        ;ENABLE RECEIVE INTERRUPT
2331  EC31  85 10  CONTIN: STA     POKMSK      ;STORE NEW VALUE TO SYSTEM MASK
2332  EC33  8D 0E D2       STA     IRQEN       ;STORE TO ACTUAL REGISTER
2333                ;
2334                ;
2335  EC36  A9 28          LDA     #$28        ;CLOCK CH.3 WITH 1.79 MHZ
2336  EC38  8D 08 D2       STA     AUDCTL      ;CLOCK CH.4 WITH CH. 3
2337                ;
2338  EC3B  A2 06          LDX     #6          ;SET PURE TONES, NO VOLUME
2339  EC3D  A9 A8          LDA     #$A8
2340  EC3F  A4 41          LDY     SOUNDR      ;TEST QUIET I/O FLAG
2341  EC41  D0 02          BNE     NOISE1      ;NE IS NORMAL (NOISY)
2342  EC43  A9 A0          LDA     #$A0
2343  EC45  9D 01 D2 NOISE1: STA   AUDC1,X
2344  EC48  CA             DEX
2345  EC49  CA             DEX
2346  EC4A  10 F9          BPL     NOISE1
2347                ;
2348  EC4C  A9 A0          LDA     #$A0
2349  EC4E  8D 05 D2       STA     AUDC3       ;TURN OFF SOUND ON CHANNEL 3
2350  EC51  AC 00 03       LDY     DDEVIC
2351  EC54  C0 60          CPY     #CASET
2352  EC56  F0 06          BEQ     CAS31       ;BRANCH IF CASSETTE IS DESIRED
2353  EC58  8D 01 D2       STA     AUDC1       ;OTHERWISE TURN OFF CHANNELS 1 AND 2
2354  EC5B  8D 03 D2       STA     AUDC2
2355                ;
2356                ;
2357  EC5E  60      CAS31:  RTS               ;RETURN
2358                ;
2359                ;
2360                ;
2361                ;
2362                ;
2363                ;
2364                ;
2365                ;
2366                ;
2367                ;
2368                ; DISABLE SEND AND DISABLE RECEIVE SUBROUTINES
2369                ;
2370  EC5F  EA      SENDDS: NOP
2371  EC60  A9 C7   RECVDS: LDA     #$C7        ;MASK OFF SERIAL BUS INTERRUPTS
2372  EC62  25 10           AND     POKMSK
2373  EC64  85 10           STA     POKMSK      ;STORE NEW VALUE TO SYSTEM MASK
2374  EC66  8D 0E D2        STA     IRQEN       ;STORE TO ACTUAL REGISTER
2375                ;
2376  EC69  A2 06           LDX     #6
2377  EC6B  A9 00           LDA     #0
2378  EC6D  9D 01 D2 ZERIT: STA     AUDC1,X
2379  EC70  CA              DEX
2380  EC71  CA              DEX
2381  EC72  10 F9           BPL     ZERIT       ;TURN OFF AUDIO VOLUME
2382                ;
```

```
EC74  60                        RTS                     ;RETURN
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ; SET DDEVICE TIME OUT VALUES IN Y,X SUBROUTINE
                        ;
EC75  AD 06 03          STTMOT: LDA     DTIMLO          ;GET DEVICE TIME OUT IN 1 SECOND INCR
EC78  6A                        ROR     A               ;PUT 6 HI BITS IN X, LO 2 BITS IN Y
EC79  6A                        ROR     A
EC7A  A8                        TAY                     ;TEMP SAVE
EC7B  29 3F                     AND     #$3F            ;MASK OFF 2 HI BITS
EC7D  AA                        TAX                     ;THIS IS HI BYTE OF TIME OUT
                        ;
EC7E  98                        TYA                     ;RESTORE
EC7F  6A                        ROR     A
EC80  29 C0                     AND     #$C0            ;MASK OFF ALL BUT 2 HI BITS
EC82  A8                        TAY                     ;THIS IS LO BYTE OF TIME OUT
                        ;
EC83  60                        RTS
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
EC84  0F EB             INTTBL: .WORD   ISRSIR          ;SERIAL INPUT READY
EC86  90 EA                     .WORD   ISRODN          ;OUTPUT DATA NEEDED
EC88  CF EA                     .WORD   ISRTD           ;TRANSMISSION DONE
                        ;
00EB                    SIRHI   =       ISRSIR/256      ;SERIAL INPUT READY ISR ADDRESS
000F                    SIRLO   =       (-256)*SIRHI+ISRSIR
00EA                    ODNHI   =       ISRODN/256      ;OUTPUT DATA NEEDED ISR ADDRESS
0090                    ODNLO   =       (-256)*ODNHI+ISRODN
00EA                    TDHI    =       ISRTD/256       ;TRANSMISSION DONE ISR ADDRESS
00CF                    TDLO    =       (-256)*TDHI+ISRTD
                        ;
                        ;
                        ;
                        ;
                        ; SEND A DATA FRAME TO AN INTELLIGENT PERIPHERAL SUBROUTINE
                        ;
                        ;
EC8A  A2 01             SENDIN: LDX     #$01
```

```
2437  EC8C  A0 FF        DELAYO: LDY     #$FF
2438  EC8E  88           DELAY1: DEY
2439  EC8F  D0 FD                BNE     DELAY1
2440  EC91  CA                   DEX
2441  EC92  D0 F8                BNE     DELAYO
2442                     ;
2443  EC94  20 6B EA             JSR     SEND        ;GO SEND THE DATA FRAME
2444                     ;
2445  EC97  A0 02                LDY     #CTIMLO     ;SET ACK TIME OUT
2446  EC99  A2 00                LDX     #CTIMHI
2447  EC9B  20 B9 ED     WAITER: JSR     SETVBX
2448                     ;
2449  EC9E  20 1A EA             JSR     WAIT        ;WAIT FOR ACK
2450                     ;
2451  ECA1  98                   TYA                 ;IF Y=0, A TIME OUT OR NACK OCCURED
2452                     ;
2453  ECA2  60                   RTS                 ;RETURN
2454                     ;
2455                     ;
2456                     ;
2457                     ;
2458                     ;
2459                     ;
2460                     ;
2461                     ;
2462                     ;
2463                     ;
2464                     ;
2465                     ; COMPUTE VALUE FOR POKEY FREQ REGS FOR THE BAUD RATE AS
2466                     ; MEASURED BY AN INTERVAL OF THE 'VCOUNT' TIMER.
2467                     ;
2468  ECA3  8D 10 03     COMPUT: STA     TIMER2
2469  ECA6  8C 11 03             STY     TIMER2+1    ;SAVE FINAL TIMER VALUE
2470  ECA9  20 04 ED             JSR     ADJUST      ;ADJUST VCOUNT VALUE
2471  ECAC  8D 10 03             STA     TIMER2      ;SAVE ADJUSTED VALUE
2472  ECAF  AD 0C 03             LDA     TIMER1
2473  ECB2  20 04 ED             JSR     ADJUST      ;ADJUST
2474  ECB5  8D 0C 03             STA     TIMER1      ;SAVE ADJUSTED TIMER1 VALUE
2475  ECB8  AD 10 03             LDA     TIMER2
2476  ECBB  38                   SEC
2477  ECBC  ED 0C 03             SBC     TIMER1
2478  ECBF  8D 12 03             STA     TEMP1       ;FIND VCOUNT DIFFERENCE
2479  ECC2  AD 11 03             LDA     TIMER2+1
2480  ECC5  38                   SEC
2481  ECC6  ED 0D 03             SBC     TIMER1+1
2482  ECC9  A8                   TAY                 ;FIND VBLANK COUNT DIFFERENCE
2483                             .IF     PALFLG
2484                             LDA     #-$9C
2485                     HITIMR: CLC
2486                             ADC     #$9C
2487                             .ENDIF
2488                             .IF     PALFLG-1
2489  ECCA  A9 7D                LDA     #-$83
2490  ECCC  18           HITIMR: CLC
```

```
2491 ECCD 69 83          ADC     #$83        ;ACCUMULATE MULTIPLICATION
2492                     .ENDIF
2493 ECCF 88             DEY
2494 ECD0 10 FA          BPL     HITIMR      ;DONE?
2495 ECD2 18             CLC
2496 ECD3 6D 12 03       ADC     TEMP1       ;TOTAL VCOUNT DIFFERENCE
2497 ECD6 A8          FINDX: TAY             ;SAVE ACCUM
2498 ECD7 4A             LSR     A
2499 ECD8 4A             LSR     A
2500 ECD9 4A             LSR     A
2501 ECDA 0A             ASL     A
2502 ECDB 38             SEC
2503 ECDC E9 16          SBC     #22         ;ADJUST TABLE INDEX
2504 ECDE AA             TAX                 ;DIVIDE INTERVAL BY 4 TO GET TABLE INDEX
2505 ECDF 98             TYA                 ;RESTORE ACCUM
2506 ECE0 29 07          AND     #7
2507 ECE2 A8             TAY                 ;PULL OFF 3 LO BITS OF INTERVAL
2508 ECE3 A9 F5          LDA     #-11
2509 ECE5 18          DOINTP: CLC
2510 ECE6 69 0B          ADC     #11         ;ACCUMULATE INTERPOLATION CONSTANT
2511 ECE8 88             DEY
2512 ECE9 10 FA          BPL     DOINTP      ;INTERPOLATION CONSTANT COMPUTATION DONE?
2513                     ;
2514 ECEB A0 00       ENINTP: LDY     #0
2515 ECED 8C 0E 03       STY     ADDCOR      ;CLEAR ADDITION CORRECTION FLAG
2516 ECF0 38             SEC
2517 ECF1 E9 07          SBC     #7          ;ADJUST INTERPOLATION CONSTANT
2518 ECF3 10 03          BPL     PLUS
2519 ECF5 CE 0E 03       DEC     ADDCOR
2520 ECF8 18          PLUS:   CLC
2521 ECF9 7D D0 ED       ADC     POKTAB,X    ;ADD CONSTANT TO LO BYTE TABLE VALUE
2522 ECFC A8             TAY                 ;LO BYTE POKEY FREQ VALUE
2523 ECFD AD 0E 03       LDA     ADDCOR
2524 ED00 7D D1 ED       ADC     POKTAB+1,X  ;ADD CARRY TO HI BYTE TABLE VALUE
2525                  ; HI BYTE POKEY FREQ VALUE
2526 ED03 60             RTS
2527                     ;
2528                     ;
2529                     ;
2530                     ;       ROUTINE TO ADJUST VCOUNT VALUE
2531                     ;
2532 ED04 C9 7C       ADJUST: CMP     #$7C
2533 ED06 30 04          BMI     ADJ1        ;LARGER THAN '7C' ?
2534 ED08 38             SEC                 ;YES,
2535 ED09 E9 7C          SBC     #$7C
2536 ED0B 60             RTS
2537 ED0C 18          ADJ1:   CLC
2538                     .IF     PALFLG
2539                     ADC     #$20
2540                     .ENDIF
2541                     .IF     PALFLG-1
2542 ED0D 69 07          ADC     #$7
2543                     .ENDIF
2544 ED0F 60             RTS
```

```
2545
2546
2547
2548
2549
2550
2551
2552                    ;       INITIAL BAUD RATE MEASUREMENT -- USED TO SET THE
2553                    ;               BAUD RATE AT THE START OF A RECORD.
2554                    ;
2555                    ;               IT IS ASSUMED THAT THE FIRST TWO BYTES OF EVERY
2556                    ;  RECORD ARE 'AA' HEX.
2557                    ;
2558 ED10 A5 11        BEGIN:  LDA     BRKKEY
2559 ED12 D0 03                BNE     NTBRK2
2560 ED14 4C A0 ED             JMP     BROKE           ;JUMP IF BREAK KEY PRESSED
2561                    ;
2562 ED17 78           NTBRK2: SEI
2563                    ;
2564 ED18 AD 17 03             LDA     TIMFLG
2565 ED1B D0 02                BNE     OKTIM1          ;BRANCH IF NOT TIMED OUT
2566 ED1D F0 25                BEQ     TOUT1           ;BRANCH IF TIME OUT
2567                    ;
2568 ED1F AD 0F D2     OKTIM1: LDA     SKSTAT
2569 ED22 29 10                AND     #$10            ;READ SERIAL PORT
2570 ED24 D0 EA                BNE     BEGIN           ;START BIT?
2571 ED26 8D 16 03             STA     SAVIO           ;SAVE SER. DATA IN
2572 ED29 AE 0B D4             LDX     VCOUNT          ;READ VERTICAL LINE COUNTER
2573 ED2C A4 14                LDY     RTCLOK+2        ;READ LO BYTE OF VBLANK CLOCK
2574 ED2E 8E 0C 03             STX     TIMER1
2575 ED31 8C 0D 03             STY     TIMER1+1        ;SAVE INITIAL TIMER VALUE
2576                    ;
2577 ED34 A2 01                LDX     #1              ;SET MODE FLAG
2578 ED36 8E 15 03             STX     TEMP3
2579 ED39 A0 0A                LDY     #10             ;SET BIT COUNTER FOR 10 BITS
2580 ED3B A5 11        COUNT:  LDA     BRKKEY
2581 ED3D F0 61                BEQ     BROKE           ;BRANCH IF BREAK KEY PRESSED
2582                    ;
2583 ED3F AD 17 03             LDA     TIMFLG
2584 ED42 D0 04                BNE     OKTIMR          ;BRANCH IF NOT TIMED OUT
2585 ED44 58           TOUT1:  CLI
2586 ED45 4C 0A EB             JMP     TOUT            ;BRANCH IF TIME OUT
2587                    ;
2588 ED48 AD 0F D2     OKTIMR: LDA     SKSTAT
2589 ED4B 29 10                AND     #$10            ;READ SERIAL PORT
2590 ED4D CD 16 03             CMP     SAVIO           ;DATA IN CHANGED YET?
2591 ED50 F0 E9                BEQ     COUNT
2592 ED52 8D 16 03             STA     SAVIO           ;YES, SAVE SER. DATA IN
2593 ED55 88                   DEY                     ;DECR. BIT COUNTER
2594 ED56 D0 E3                BNE     COUNT           ;DONE?
2595                    ;
2596 ED58 CE 15 03             DEC     TEMP3           ;YES,
2597 ED5B 30 12                BMI     GOREAD          ;DONE WITH BOTH MODES?
2598 ED5D AD 0B D4             LDA     VCOUNT
```

```
2599  ED60  A4 14          LDY   RTCLOK+2    ;READ TIMER LO & HI BYTES
2600  ED62  20 A3 EC       JSR   COMPUT      ;NO, COMPUTE BAUD RATE
2601  ED65  8C EE 02       STY   CBAUDL
2602  ED68  8D EF 02       STA   CBAUDH      ;SET BAUD RATE INTO RAM CELLS
2603  ED6B  A0 09          LDY   #9          ;SET BIT COUNTER FOR 9 BITS
2604  ED6D  D0 CC          BNE   COUNT
2605                ;
2606  ED6F  AD EE 02  GOREAD: LDA   CBAUDL
2607  ED72  8D 04 D2       STA   AUDF3
2608  ED75  AD EF 02       LDA   CBAUDH
2609  ED78  8D 06 D2       STA   AUDF4       ;SET POKEY FREQ REGS FOR BAUD RATE
2610  ED7B  A9 00          LDA   #0
2611  ED7D  8D 0F D2       STA   SKSTAT
2612  ED80  AD 32 02       LDA   SSKCTL
2613  ED83  8D 0F D2       STA   SKSTAT      ;INIT. POKEY SERIAL PORT
2614  ED86  A9 55          LDA   #$55
2615  ED88  91 32          STA   (BUFRLO),Y  ;STORE '$55' AS FIRST RCV. BUFFER
2616  ED8A  C8             INY
2617  ED8B  91 32          STA   (BUFRLO),Y
2618  ED8D  A9 AA          LDA   #$AA
2619  ED8F  85 31          STA   CHKSUM      ;STORE CHECKSUM FOR 2 BYTES OF '$AA'
2620  ED91  18             CLC
2621  ED92  A5 32          LDA   BUFRLO
2622  ED94  69 02          ADC   #2
2623  ED96  85 32          STA   BUFRLO
2624  ED98  A5 33          LDA   BUFRHI
2625  ED9A  69 00          ADC   #0
2626  ED9C  85 33          STA   BUFRHI      ;INCR. BUFFER POINTER BY 1
2627  ED9E  58             CLI
2628  ED9F  60             RTS
2629                ;
2630                ;
2631                ;
2632  EDA0  20 5F EC  BROKE:  JSR   SENDDS      ;BREAK KEY WAS PRESSED, SO PREPARE
2633  EDA3  A9 3C          LDA   #MOTRST     ;TO RETURN
2634  EDA5  8D 02 D3       STA   PACTL       ;TURN OFF MOTOR
2635  EDA8  8D 03 D3       STA   PBCTL       ;RAISE NOT COMMAND LINE
2636                ;
2637  EDAB  A9 80          LDA   #BRKABT
2638  EDAD  85 30          STA   STATUS      ;STORE BREAK ABORT STATUS CODE
2639                ;
2640  EDAF  AE 18 03       LDX   STACKP
2641  EDB2  9A             TXS               ;RESTORE STACK POINTER
2642                ;
2643  EDB3  C6 11          DEC   BRKKEY      ;SET BREAK KEY FLAG TO NONZERO
2644  EDB5  58             CLI               ;ALLOW IRQ'S
2645                ;
2646  EDB6  4C 0D EA       JMP   RETURN      ;GO RETURN
2647                ;
2648                ;
2649                ;
2650                ;
2651                ;
2652  EDB9  A9 EC  SETVBX: LDA   #JTADRL     ;STORE TIME OUT ROUTINE ADDRESS
```

```
2653  EDBB  8D 26 02          STA     CDTMA1
2654  EDBE  A9 EB             LDA     #JTADRH
2655  EDC0  8D 27 02          STA     CDTMA1+1
2656                ;
2657  EDC3  A9 01              LDA     #1              ;SET FOR TIMER 1
2658                ;
2659  EDC5  78                 SEI                     ;THE SETVBL ROUTINE NEEDS THIS TO CUT SHORT
2660  EDC6  20 5C E4           JSR     SETVBV          ;ANY VBLANKS THAT OCCUR
2661  EDC9  A9 01              LDA     #1              ;SET FOR TIMER 1
2662  EDCB  8D 17 03           STA     TIMFLG          ;SET FLAG TO NOT TIMED OUT
2663  EDCE  58                 CLI
2664  EDCF  60                 RTS
2665                ;
2666                ;
2667                ;
2668                ;
2669                ;
2670                ;
2671                ;
2672                ; 'VCOUNT' INTERVAL TIMER MEASUREMENT -- TO -- POKEY FREQ REG VALUE
2673                ;               CONVERSION TABLE
2674                ;
2675                ;
2676                ; THE VALUES STORED IN THE TABLE ARE 'AUDF+7'.
2677                ;
2678                ;       THE FOLLOWING FORMULAS WERE USED TO DETERMINE THE TABLE VALUES:
2679                ;
2680                ;               F OUT= F IN/(2*(AUDF+M)) , WHERE F IN=1.78979 MHZ. & M=7
2681                ;
2682                ;        FROM THIS WAS DERIVED THE FORMULA USED TO COMPUTE THE
2683                ;        TABLE VALUES BASED ON A MEASUREMENT OF THE PERIOD BY
2684                ;        AN INTERVAL OF THE 'VCOUNT' TIMER.
2685                ;
2686                ;               AUDF+7=(11.365167)*T OUT, WHERE T OUT=# OF COUNTS
2687                ;               (127 USEC. RESOLUTION) OF 'VCOUNT' FOR 1
2688                ;               CHARACTER TIME (10 BIT TIMES).
2689                ;
2690                ;
2691                ;
2692                ;
2693                ;                       AUDF+7          BAUD RATE       VCOUNT INTERVAL
2694                ;                       ------          ---------       ---------------
2695                ;       .WORD   $27C            ;1407           56
2696                ;       .WORD   $2D7            ;1231           64
2697                ;       .WORD   $332            ;1094           72
2698                ;       .WORD   $38D            ;985            80
2699  EDD0  E8 03   POKTAB: .WORD   $3E8            ;895            88
2700  EDD2  43 04           .WORD   $443            ;820            96
2701  EDD4  9E 04           .WORD   $49E            ;757            104
2702  EDD6  F9 04           .WORD   $4F9            ;703            112
2703  EDD8  54 05           .WORD   $554            ;656            120
2704  EDDA  AF 05           .WORD   $5AF            ;615            128
2705  EDDC  0A 06           .WORD   $60A            ;579            136
2706  EDDE  65 06           .WORD   $665            ;547            144
```

```
EDE0  C0 06                    .WORD   $6C0          ;518          152
EDE2  1A 07                    .WORD   $71A          ;492          160
EDE4  75 07                    .WORD   $775          ;469          168
EDE6  D0 07                    .WORD   $7D0          ;447          176
                        ;      .WORD   $82B          ;428          184
                        ;      .WORD   $886          ;410          192
                        ;      .WORD   $8E1          ;394          200
                        ;      .WORD   $93C          ;379          208
                        ;      .WORD   $997          ;365          216
                        ;      .WORD   $9F2          ;352          224
                        ;      .WORD   $A4D          ;339          232
                        ;      .WORD   $AA8          ;328          240
                        ;      .WORD   $B03          ;318          248
                        ;
                        ;
                        ;
                        ;
                        ;*****************************************************************
EDE8                    CRNTP3  =*
                                *=$14
0014  02                SIOSPR: .BYTE   DSKORG-CRNTP3 ;^GSIOL IS TOO LONG
```

```
2728
2729                            .TITLE  'DISK  ***** DISKP.SRC ***** 3/9/79 *****  4:00:00 P.M. '
2730                    ;
2731                    ;
2732                    ;
2733                    ;
2734                    ;
2735                    ;
2736  0002              STATVH  =       DVSTAT/256
2737  00EA              STATVL  =       (-256)*STATVH+DVSTAT ;STATUS POINTER
2738                    ;
2739                    ;
2740                    ;
2741                    ;
2742                    ;       CONSTANT EQUATES
2743                    ;
2744  0031              DISKID  =       $31             ;SERIAL BUS DISK I.D.
2745  0050              PUTSEC  =       $50             ;DISK PUT SECTOR DCB COMMAND
2746                    ; READ  =       $52             ;DISK GET SECTOR DCB COMMAND
2747                    ; WRITE =       $57             ;DISK PUT SECTOR WITH READ CHECK DCB COMMAND
2748  0053              STATC   =       $53             ;DISK STATUS DCB COMMAND
2749  0021              FOMAT   =       $21             ;DISK FORMAT DCB COMMAND !!!!! *****
2750  0000              NODAT   =       0               ;SIO COMMAND FOR "NO DATA" OPERATION
2751  0040              GETDAT  =       $40             ;SIO COMMAND FOR "DATA FROM DEVICE"
2752  0080              PUTDAT  =       $80             ;SIO COMMAND FOR "DATA TO DEVICE"
2753                    ;
2754                    ;
2755                    ;       VECTORS
2756                    ;
2757                            *=$E450
2758                    ;
2759  E450  4C EA ED            JMP     DINIT           ;DISK INIT. VECTOR
2760  E453  4C F0 ED            JMP     DSKIF           ;DISK INTERFACE ENTRY POINT
2761                    ;
2762                    ;
2763                    ;
2764                    ;
2765                    ;
2766                    ;
2767                    ;       CONSTANTS
2768                    ;
2769                            *=DSKORG
2770                    ;
2771                    ;
2772                    ;
2773                    ;
2774                    ;
2775                    ;
2776                    ;
2777                    ;
2778                    ;
2779                    ;*************************************************************************
2780                    ;       DISK INTERFACE ROUTINE STARTS HERE
2781                    ;*************************************************************************
```

```
2782                                ;
2783                                ;
2784                                ;
2785                                ;
2786                                ;       DISK INTERFACE INITIALIZATION ROUTINE
2787                                ;
2788  EDEA  A9 A0           DINIT:  LDA     #160
2789  EDEC  8D 46 02                STA     DSKTIM      ;SET INITIAL DISK TIMEOUT TO 160 SEC
2790  EDEF  60                      RTS
2791                                ;
2792                                ;
2793                                ;
2794                                ;       DISK INTERFACE ENTRY POINT
2795                                ;
2796  EDF0  A9 31           DSKIF:  LDA     #DISKID
2797  EDF2  8D 00 03                STA     DDEVIC      ;SET SERIAL BUS I.D IN DCB
2798  EDF5  AD 46 02                LDA     DSKTIM
2799  EDF8  AE 02 03                LDX     DCOMND
2800  EDFB  E0 21                   CPX     #FOMAT      ;IS COMMAND A FORMAT COMMAND?
2801  EDFD  F0 02                   BEQ     PUTDTO
2802  EDFF  A9 07                   LDA     #7          ;NO, SET TIMEOUT TO 7 SECS.
2803  EE01  8D 06 03       PUTDTO: STA     DTIMLO      ;PUT DISK TIMEOUT IN DCB
2804  EE04  A2 40                   LDX     #GETDAT     ;SET "GET DATA" COMMAND FOR SIO
2805  EE06  A0 80                   LDY     #$80        ;SET BYTE COUNT TO 128
2806  EE08  AD 02 03                LDA     DCOMND      ;READ COMMAND IN DCB
2807  EE0B  C9 57                   CMP     #WRITE      ;IS COMMAND A "PUT SECTOR" COMMAND?
2808  EE0D  D0 02                   BNE     CKSTC
2809  EE0F  A2 80                   LDX     #PUTDAT     ;YES, SET "PUT DATA" COMMAND FOR SIO
2810  EE11  C9 53           CKSTC:  CMP     #STATC      ;IS COMMAND A STATUS COMMAND?
2811  EE13  D0 0C                   BNE     PUTCNT
2812  EE15  A9 EA                   LDA     #STATVL
2813  EE17  8D 04 03                STA     DBUFLO
2814  EE1A  A9 02                   LDA     #STATVH
2815  EE1C  8D 05 03                STA     DBUFHI      ;SET BUFFER ADDR TO GLOBAL STATUS BUFFER
2816  EE1F  A0 04                   LDY     #4          ;YES, SET BYTE COUNT TO 4
2817  EE21  8E 03 03       PUTCNT: STX     DSTATS      ;PUT STATUS COMMAND FOR SIO IN DCB
2818  EE24  8C 08 03                STY     DBYTLO
2819  EE27  A9 00                   LDA     #0
2820  EE29  8D 09 03                STA     DBYTHI      ;PUT BYTE COUNT IN DCB
2821  EE2C  20 59 E4                JSR     SIOV        ;CALL SERIAL I/O.
2822  EE2F  10 01                   BPL     GOODST      ;NO ERROR
2823  EE31  60                      RTS                 ;NO, GO BACK
2824  EE32  AD 02 03        GOODST: LDA     DCOMND      ;READ THE COMMAND
2825  EE35  C9 53                   CMP     #STATC      ;WAS IT A STATUS COMMAND?
2826  EE37  D0 0A                   BNE     PUTBC
2827  EE39  20 6D EE                JSR     PUTADR      ;PUT BUFFER ADDR IN TEMP REG.
2828  EE3C  A0 02                   LDY     #2
2829  EE3E  B1 15                   LDA     (BUFADR),Y  ;READ DISK TIMEOUT VALUE BYTE OF STATUS
2830  EE40  8D 46 02                STA     DSKTIM      ;PUT IT IN DISK TIMEOUT REG.
2831  EE43  AD 02 03        PUTBC:  LDA     DCOMND
2832  EE46  C9 21                   CMP     #FOMAT      ;WAS COMMAND A FORMAT COMMAND?
2833  EE48  D0 1F                   BNE     ENDDIF
2834  EE4A  20 6D EE        FMTD:   JSR     PUTADR      ;YES, PUT BUFFER ADDR INTO TEMP REG
2835  EE4D  A0 FE                   LDY     #$FE        ;SET BUFFER POINTER
```

```
2836  EE4F  C8          TWICE:  INY
2837  EE50  C8                  INY                 ;INCR BUFFER POINTER BY 2
2838  EE51  B1 15        RDBAD:  LDA     (BUFADR),Y  ;READ LO BYTE BAD SECTOR DATA
2839  EE53  C9 FF               CMP     #$FF
2840  EE55  D0 F8               BNE     TWICE       ;IS IT "FF" ?
2841  EE57  C8                  INY                 ;YES,
2842  EE58  B1 15               LDA     (BUFADR),Y  ;READ HI BYTE BAD SECTOR DATA
2843  EE5A  C8                  INY
2844  EE5B  C9 FF               CMP     #$FF
2845  EE5D  D0 F2               BNE     RDBAD       ;IS IT "FF" ?
2846  EE5F  88                  DEY
2847  EE60  88                  DEY                 ;YES,
2848  EE61  8C 08 03            STY     DBYTLO      ;PUT BAD SECTOR BYTE COUNT INTO DCB
2849  EE64  A9 00               LDA     #0
2850  EE66  8D 09 03            STA     DBYTHI
2851  EE69  AC 03 03     ENDDIF: LDY     DSTATS
2852  EE6C  60                  RTS
2853                     ;
2854                     ;
2855                     ;
2856                     ;
2857                     ;       S U B R O U T I N E S
2858                     ;
2859                     ;
2860                     ;       PUT BUFFER ADDR FROM DCB INTO TEMP REG
2861                     ;
2862  EE6D  AD 04 03     PUTADR: LDA     DBUFLO
2863  EE70  85 15               STA     BUFADR
2864  EE72  AD 05 03            LDA     DBUFHI
2865  EE75  85 16               STA     BUFADR+1    ;PUT BUFFER ADDR IN TEMP REG
2866  EE77  60                  RTS
2867                     ;*****************************************************************
2868                     ;
2869                     ;
2870                     ;       SPARE BYTE OR MODULE TOO LONG FLAG
2871                     ;
2872  EE78               CRNTP4  =       *
2873                     ;
2874                             *=$14
2875  0014  00           DSKSPR: .BYTE   PRNORG-CRNTP4 ;^GDISKP TOO LONG
2876                     ;
```

```
2877                                    .PAGE
2878                                    .TITLE  'PRINTER  ***** PRINTP.SRC ***** 3/9/79 *****  4:00:00 P
2879                            ;
2880                            ;
2881                            ;
2882                            ;
2883                            ;
2884                            ;
2885                            ;
2886                            ;
2887                            ;
2888                            ;
2889                            ;
2890                            ;       DEVICE NUMBER OR CODE EQUATES
2891                            ;
2892  0002                      OPNOUT  =       $2              ;IOCB OPEN FOR OUTPUT COMMAND
2893  0028                      NBUFSZ  =       40              ;PRINT NORMAL BUFFER SIZE
2894  0014                      DBUFSZ  =       20              ;PRINT DOUBLE BUFFFER SIZE
2895  001D                      SBUFSZ  =       29              ;PRINT SIDEWAYS BUFFER SIZE
2896  0040                      PDEVN   =       $40             ;PRINTER DEVICE NUMBER
2897                            ; STATC =       $53             ;DCB STATUS COMMAND CODE
2898  0057                      WRITEC  =       $57             ;DCB WRITE COMMAND
2899  0020                      SPACE   =       $20             ;ASCII SPACE CHAR.
2900  004E                      N       =       $4E             ;ASCII "N" CHAR.
2901  0044                      D       =       $44             ;ASCII "D" CHAR.
2902  0053                      S       =       $53             ;ASCII "S" CHAR.
2903                            ;
2904                            ;
2905                            ;       PRINTER HANDLER ENTRY POINTS
2906                            ;
2907                            ;
2908                            ;
2909                            ;
2910                                    *=$E430
2911                            ;
2912  E430  9E EE                       .WORD   PHOPEN-1        ;PRINTER HANDLER OPEN
2913  E432  DB EE                       .WORD   PHCLOS-1        ;PH CLOSE
2914  E434  9D EE                       .WORD   BADST-1         ;PH READ
2915  E436  A6 EE                       .WORD   PHWRIT-1        ;PH WRITE
2916  E438  80 EE                       .WORD   PHSTAT-1        ;PH STATUS
2917  E43A  9D EE                       .WORD   BADST-1         ;PH SPECIAL
2918  E43C  4C 78 EE                    JMP     PHINIT          ;PH INIT.
2919  E43F  00                          .BYTE   0               ;ROM FILLER
2920                            ;
2921                            ;
2922                            ;
2923                            ;
2924                            ;
2925                                    *=PRNORG
2926                            ;
2927                            ;
2928                            ;
2929                            ;
2930                            ;       PRINTER HANDLER INITIALIZATION ROUTINE
```

```
2931                            ;
2932  EE78  A9 1E              PHINIT: LDA     #30
2933  EE7A  85 1C                      STA     PTIMOT      ;SET UP INITIAL PRINTER TIMEOUT OF 30 SEC.
2934  EE7C  60                         RTS
2935                            ;
2936                            ;
2937                            ;       PRINTER HANDLER CONSTANTS
2938                            ;
2939  EE7D  EA 02               PHSTLO: .WORD   DVSTAT      ;STATUS BUFFER POINTER
2940  EE7F  C0 03               PHCHLO: .WORD   PRNBUF      ;CHAR. BUFFER POINTER
2941                            ;
2942                            ;
2943                            ;
2944                            ;       ************************
2945                            ;       PRINTER HANDLER ROUTINES
2946                            ;       ************************
2947                            ;
2948                            ;
2949                            ;
2950                            ;
2951                            ;
2952                            ;       PRINTER HANDLER STATUS ROUTINE
2953                            ;
2954  EE81  A9 04               PHSTAT: LDA     #4
2955  EE83  85 1E                       STA     PBUFSZ      ;SET BUFFER SIZE TO 4 BYTES
2956  EE85  AE 7D EE                    LDX     PHSTLO
2957  EE88  AC 7E EE                    LDY     PHSTLO+1    ;SET POINTER TO STATUS BUFFER
2958  EE8B  A9 53                       LDA     #STATC      ;SET COMMAND TO "STATUS"
2959  EE8D  8D 02 03                    STA     DCOMND      ;SET STATUS COMMAND
2960  EE90  8D 0A 03                    STA     DAUX1
2961  EE93  20 E6 EE                    JSR     SETDCB      ;GO SETUP DCB
2962  EE96  20 59 E4                    JSR     SIOV        ;SEND STATUS COMMAND
2963  EE99  30 03                       BMI     BADST       ;GO IF ERROR
2964  EE9B  20 14 EF                    JSR     PHPUT       ;YES, PUT STATUS INTO GLOBAL BUFFER.
2965  EE9E  60                  BADST:  RTS
2966                            ;
2967                            ;
2968                            ;
2969                            ;
2970                            ;       PRINTER HANDLER OPEN ROUTINE
2971                            ;
2972  EE9F  20 81 EE            PHOPEN: JSR     PHSTAT      ;DO STATUS COMMAND TO SIO
2973  EEA2  A9 00                       LDA     #0
2974  EEA4  85 1D                       STA     PBPNT       ;CLEAR PRINT BUFFER POINTER
2975  EEA6  60                          RTS
2976                            ;
2977                            ;
2978                            ;
2979                            ;
2980                            ;       PRINTER HANDLER WRITE ROUTINE
2981                            ;
2982  EEA7  85 1F               PHWRIT: STA     PTEMP       ;SAVE ACCUM
2983  EEA9  20 1A EF                    JSR     PRMODE      ;GO DETERMINE PRINT MODE
2984  EEAC  A6 1D                       LDX     PBPNT
```

```
2985 EEAE A5 1F          LDA   PTEMP       ;GET CHAR. SENT BY CIO
2986 EEB0 9D C0 03       STA   PRNBUF,X    ;PUT CHAR. IN PRINT BUFFER
2987 EEB3 E8             INX               ;INCR. BUFFER POINTER
2988 EEB4 E4 1E          CPX   PBUFSZ      ;BUFFER POINTER=BUFFER SIZE?
2989 EEB6 F0 13          BEQ   BUFFUL
2990 EEB8 86 1D          STX   PBPNT       ;SAVE BUFFER POINTER
2991 EEBA C9 9B          CMP   #CR         ;IS CHAR. = EOL ?
2992 EEBC F0 03          BEQ   BLFILL      ;IF YES, GO DO BLANK FILL.
2993 EEBE A0 01          LDY   #SUCCES     ;PUT GOOD STATUS IN Y REG FOR CIO.
2994 EEC0 60             RTS
2995 EEC1 A9 20    BLFILL: LDA   #SPACE     ;PUT BLANK IN ACCUM.
2996 EEC3 9D C0 03 FILLBF: STA   PRNBUF,X    ;STORE IT IN PRINT BUFFER.
2997 EEC6 E8             INX
2998 EEC7 E4 1E          CPX   PBUFSZ
2999 EEC9 D0 F8          BNE   FILLBF      ;BUFFER BLANK FILLED?
3000 EECB A9 00    BUFFUL: LDA   #0
3001 EECD 85 1D          STA   PBPNT       ;CLEAR PRINT BUFFER POINTER
3002 EECF AE 7F EE       LDX   PHCHLO
3003 EED2 AC 80 EE       LDY   PHCHLO+1    ;SET POINTER TO PRINT BUFFER
3004 EED5 20 E6 EE       JSR   SETDCB      ;GO SETUP DCB
3005 EED8 20 59 E4       JSR   SIOV        ;SEND PRINT COMMAND
3006 EEDB 60             RTS               ;YES.
3007
3008
3009
3010
3011                     ;       PRINTER HANDLER CLOSE ROUTINE
3012                     ;
3013 EEDC 20 1A EF PHCLOS: JSR   PRMODE     ;GO DETERMINE PRINT MODE
3014 EEDF A6 1D          LDX   PBPNT
3015 EEE1 D0 DE          BNE   BLFILL
3016 EEE3 A0 01          LDY   #SUCCES
3017 EEE5 60             RTS
3018
3019
3020
3021
3022
3023
3024
3025
3026                     ;       S U B R O U T I N E S
3027                     ;
3028                     ;
3029                     ;
3030                     ;
3031                     ;
3032                     ;       SET UP DCB TO CALL SIO
3033                     ;
3034 EEE6 8E 04 03 SETDCB: STX   DBUFLO
3035 EEE9 8C 05 03       STY   DBUFHI      ;SET BUFFER POINTER
3036 EEEC A9 40          LDA   #PDEVN
3037 EEEE 8D 00 03       STA   DDEVIC      ;SET PRINTER BUS I.D. FOR DCB
3038 EEF1 A9 01          LDA   #1
```

```
3039  EEF3  8D 01 03          STA     DUNIT       ;SET UNIT NUMBER TO 1
3040  EEF6  A9 80             LDA     #$80        ;DEVICE WILL EXPECT DATA
3041  EEF8  AE 02 03          LDX     DCOMND
3042  EEFB  E0 53             CPX     #STATC      ;STATUS COMMAND?
3043  EEFD  D0 02             BNE     PSIOC
3044  EEFF  A9 40             LDA     #$40        ;EXPECT DATA FROM DEVICE
3045  EF01  8D 03 03    PSIOC: STA     DSTATS      ;SET SIO MODE COMMAND.
3046  EF04  A5 1E             LDA     PBUFSZ
3047  EF06  8D 08 03          STA     DBYTLO      ;SET LO BYTE COUNT
3048  EF09  A9 00             LDA     #0
3049  EF0B  8D 09 03          STA     DBYTHI      ;SET HI BYTE COUNT
3050  EF0E  A5 1C             LDA     PTIMOT
3051  EF10  8D 06 03          STA     DTIMLO      ;SET DEVICE TIMEOUT COUNT
3052  EF13  60                RTS
3053                    ;
3054                    ;
3055                    ;
3056                    ;
3057                    ; GET DEVICE TIMEOUT FROM STATUS & SAVE IT
3058                    ;
3059  EF14  AD EC 02    PHPUT: LDA     DVSTAT+2
3060  EF17  85 1C             STA     PTIMOT      ;SAVE DEVICE TIMEOUT
3061  EF19  60                RTS
3062                    ;
3063                    ;
3064                    ;
3065                    ;
3066                    ; DETERMINE PRINT MODE & SETUP PRINT BUFFER SIZE, DCB PRINT
3067                    ; COMMAND, & DCB AUX1 FOR PRINT MODE
3068                    ;
3069  EF1A  A0 57       PRMODE: LDY     #WRITEC     ;PUT WRITE COMMAND IN Y REG
3070  EF1C  A5 2B             LDA     ICAX2Z      ;READ PRINT MODE
3071  EF1E  C9 4E       CMODE: CMP     #N
3072  EF20  D0 04             BNE     CDUBL       ;PRINT NORMAL ?
3073  EF22  A2 28             LDX     #NBUFSZ     ;YES, SET NORMAL CHAR. BUFFER SIZE
3074  EF24  D0 0E             BNE     SETBSZ
3075  EF26  C9 44       CDUBL: CMP     #D
3076  EF28  D0 04             BNE     CSIDE       ;PRINT DOUBLE?
3077  EF2A  A2 14             LDX     #DBUFSZ     ;YES, SET DOUBLE CHAR. BUFFER SIZE
3078  EF2C  D0 06             BNE     SETBSZ
3079  EF2E  C9 53       CSIDE: CMP     #S          ;PRINT SIDEWAYS ?
3080  EF30  D0 0B             BNE     GOERR       ;IF NOT, GO TO ERROR ROUTINE
3081  EF32  A2 1D             LDX     #SBUFSZ     ;YES, SET SIDEWAYS BUFFER SIZE
3082  EF34  86 1E       SETBSZ: STX     PBUFSZ      ;STORE PRINT BUFFER SIZE
3083  EF36  8C 02 03          STY     DCOMND      ;STORE DCB COMMAND
3084  EF39  8D 0A 03          STA     DAUX1       ;STORE DCB AUX1 PRINT MODE
3085  EF3C  60                RTS
3086  EF3D  A9 4E       GOERR: LDA     #N          ;SET DEFAULT PRINT MODE TO NORMAL
3087  EF3F  D0 DD             BNE     CMODE
3088                    ;*****************************************************************
3089                    ;
3090                    ;
3091                    ; SPARE BYTE OR MODULE TOO LONG FLAG
3092                    ;
```

```
3093  EF41                    CRNTP5  =       *
3094                          ;
3095                                  *=$14
3096                          ;
3097  0014  00                PRNSPR: .BYTE   CASORG-CRNTP5 ;^GPRINTP TOO LONG
3098                          ;
```

```
3099                                    .PAGE
3100                                    .TITLE  'CASSET HANDLER 3/12 (DK1:CASCV)'
3101  0003                      CBUFH   =       CASBUF/256
3102  00FD                      CBUFL   =       (-256)*CBUFH+CASBUF
3103  0040                      SRSTA   =       $40             ;SIO READ STATUS
3104  0080                      SWSTA   =       $80             ;SIO WRITE STATUS
3105                            ;MOTRGO =       $34
3106                            ;MOTRST =       $3C
3107                            ;
3108                            ;
3109  00FC                      DTA     =       $FC             ;DATA RECORD TYPE BYTE
3110  00FA                      DT1     =       $FA             ;LAST DATA RECORD
3111  00FE                      EOT     =       $FE             ;END OF TAPE
3112  00FB                      HDR     =       $FB             ;HEADER
3113  0002                      TONE1   =       2               ;CHANGE TO RECORD MODE TONE
3114  0001                      TONE2   =       1               ;PRESS PLAY TONE
3115                            ;
3116                            ;
3117                            ;
3118                                    *=CASETV
3119  E440  4B EF 2A F0                 .WORD   OPENC-1,CLOSEC-1,GBYTE-1,PBYTE-1,STATU-1,SPECIAL-1
3120  E444  D5 EF 0F F0
3121  E448  27 F0 4A EF
3122  E44C  4C 41 EF                    JMP     INIT
3123  E44F  00                          .BYTE   0               ;ROM FILLER BYTE
3124                            ;
3125                            ;
3126                            ;
3127                            ; USED IN MONITP FOR CASSETTE BOOT
3128                            ;
3129                                    *=RBLOKV
3130  E47A  4C E9 EF                    JMP     RBLOK
3131                            ;
3132                                    *=CSOPIV
3133  E47D  4C 5D EF                    JMP     OPINP
3134                            ;
3135                            ;
3136                                    *=CASORG
3137                            ;
3138                            ;
3139                            ; INIT ROUTINE
3140                            ;
3141  EF41  A9 CC               INIT:   LDA     #$CC
3142  EF43  8D EE 02                    STA     CBAUDL
3143  EF46  A9 05                       LDA     #$05
3144  EF48  8D EF 02                    STA     CBAUDH          ;SET CASSET BAUD RATE TO 600
3145                            SPECIAL:                        ;THATS ALL FOLKS
3146  EF4B  60                          RTS
```

```
3147                                        .PAGE
3148                                ;
3149                                ; OPEN FUNCTION - WITH NO TIMING ADJUST
3150                                ;
3151  EF4C  A5 2B           OPENC: LDA     ICAX2Z       ;GET AX2
3152  EF4E  85 3E                  STA     FTYPE        ;SAVE IT FOR FUTURE REFERENCE
3153  EF50  A5 2A                  LDA     ICAX1Z
3154  EF52  29 0C                  AND     #$0C         ;IN AND OUT BITS
3155  EF54  C9 04                  CMP     #$04
3156  EF56  F0 05                  BEQ     OPINP
3157  EF58  C9 08                  CMP     #$08         ;SEE IF OPEN FOR OUTPUT
3158  EF5A  F0 39                  BEQ     OPOUT
3159  EF5C  60                     RTS                  ;IF ALREADY OPEN, RETURN LEAVING STATUS=$84
3160  EF5D  A9 00           OPINP: LDA     #0
3161  EF5F  8D 89 02               STA     WMODE        ;SET READ MODE
3162  EF62  85 3F                  STA     FEOF         ;NO EOF YET
3163  EF64  A9 01           SFH:   LDA     #TONE2       ;TONE FOR PRESS PLAY
3164  EF66  20 58 F0               JSR     BEEP         ;GO BEEP
3165  EF69  30 24                  BMI     OPNRTN       ;IF ERROR DURING BEEP
3166  EF6B  A9 34                  LDA     #MOTRGO
3167  EF6D  8D 02 D3               STA     PACTL        ;TURN MOTOR ON
3168                                .IF     PALFLG
3169                                LDY     #$E0
3170                                LDX     #1
3171                                .ENDIF
3172                                .IF     PALFLG-1
3173  EF70  A0 40                  LDY     #$40         ;5-31-79 9 SEC READ LEADER
3174  EF72  A2 02                  LDX     #2
3175                                .ENDIF
3176  EF74  A9 03                  LDA     #3
3177  EF76  8D 2A 02               STA     CDTMF3
3178  EF79  20 5C E4               JSR     SETVBV       ;SET UP VBLANK TIMER
3179  EF7C  AD 2A 02        WAITTM: LDA     CDTMF3
3180  EF7F  D0 FB                  BNE     WAITTM       ;WAIT FOR MOTOR TO COME UP TO SPEED
3181  EF81  A9 80                  LDA     #$80         ;NEXT BYTE=NO BYTES IN BUFFER
3182  EF83  85 3D                  STA     BPTR
3183  EF85  8D 8A 02               STA     BLIM
3184  EF88  4C D3 EF               JMP     OPOK         ;OPEN OK
3185                                ;
3186                                ; OPEN FOR OUTPUT
3187                                ;
3188  EF8B  A0 80           PBRK:  LDY     #BRKABT      ;BREAK KEY ABORT STATUS
3189  EF8D  C6 11                  DEC     BRKKEY       ;RESET BREAK KEY
3190  EF8F  A9 00           OPNRTN: LDA    #0           ;CLEAR WRITE MODE FLAG
3191  EF91  8D 89 02               STA     WMODE
3192  EF94  60                     RTS                  ;AND EXIT.
3193                                ;
3194  EF95  A9 80           OPOUT: LDA     #$80
3195  EF97  8D 89 02               STA     WMODE        ;SET WRITE MODE
3196  EF9A  A9 02                  LDA     #TONE1       ;TELL USER TO TURN ON RECORD MODE
3197  EF9C  20 58 F0               JSR     BEEP
3198  EF9F  30 EE                  BMI     OPNRTN       ;IF ERROR DURING BEEP
3199  EFA1  A9 CC                  LDA     #$CC         ;SET BAUD RATE
3200  EFA3  8D 04 D2               STA     AUDF3        ;WHICH SEEMS TO BE NESSECARY
```

```
3201  EFA6  A9 05          LDA     #$05        ;FOR SOME OBSCURE REASON
3202  EFA8  8D 06 D2       STA     AUDF4
3203  EFAB  A9 60          LDA     #$60
3204  EFAD  8D 00 03       STA     DDEVIC
3205  EFB0  20 68 E4       JSR     SENDEV      ;TELL POKEY TO WRITE MARKS
3206  EFB3  A9 34          LDA     #MOTRGO     ;WRITE 5 SEC BLANK TAPE
3207  EFB5  8D 02 D3       STA     PACTL
3208  EFB8  A9 03          LDA     #3
3209                       .IF     PALFLG
3210                       LDX     #$3
3211                       LDY     #$C0
3212                       .ENDIF
3213                       .IF     PALFLG-1
3214  EFBA  A2 04          LDX     #4          ;5/30/79 20 SEC LEADER
3215  EFBC  A0 80          LDY     #$80
3216                       .ENDIF
3217  EFBE  20 5C E4       JSR     SETVBV
3218  EFC1  A9 FF          LDA     #$FF
3219  EFC3  8D 2A 02       STA     CDTMF3
3220  EFC6  A5 11       WDLR: LDA  BRKKEY
3221  EFC8  F0 C1          BEQ     PBRK        ;IF BREAK DURING WRITE LEADER
3222  EFCA  AD 2A 02       LDA     CDTMF3
3223  EFCD  D0 F7          BNE     WDLR
3224  EFCF  A9 00          LDA     #0          ;INIT BUFFER POINTER
3225  EFD1  85 3D          STA     BPTR
3226  EFD3  A0 01       OPOK: LDY  #SUCCES
3227  EFD5  60             RTS
```

```
3228                            .PAGE
3229                    ;
3230                    ; GET BYTE
3231                    ;
3232  EFD6  A5 3F       GBYTE:  LDA     FEOF            ;IF AT EOF ALREADY
3233  EFD8  30 33               BMI     ISEOF           ;RETURN EOF STATUS
3234  EFDA  A6 3D               LDX     BPTR            ;BUFFER POINTER
3235  EFDC  EC 8A 02            CPX     BLIM            ;IF END OF BUFFER
3236  EFDF  F0 08               BEQ     RBLOK           ;READ ANOTHER BLOCK
3237  EFE1  BD 00 04            LDA     CASBUF+3,X      ;GET NEXT BYTE
3238  EFE4  E6 3D               INC     BPTR            ;BUMP POINTER
3239  EFE6  A0 01               LDY     #SUCCES         ;OK STATUS
3240  EFE8  60          GBX:    RTS
3241  EFE9  A9 52       RBLOK:  LDA     #'R             ;READ OPCODE
3242  EFEB  20 95 F0            JSR     SIOSB           ;SIO ON SYS BUF
3243  EFEE  98                  TYA
3244  EFEF  30 F7               BMI     GBX             ;IF SIO ERRORS, RETURN
3245  EFF1  A9 00               LDA     #0
3246  EFF3  85 3D               STA     BPTR            ;RESET POINTER
3247  EFF5  A2 80               LDX     #$80            ;DEFAULT # BYTES
3248  EFF7  AD FF 03            LDA     CASBUF+2
3249  EFFA  C9 FE               CMP     #EOT
3250  EFFC  F0 0D               BEQ     ATEOF           ;IF HEADER, GO READ AGAIN
3251  EFFE  C9 FA               CMP     #DT1            ;IF LAST DATA REC
3252  F000  D0 03               BNE     NLR
3253  F002  AE 7F 04            LDX     CASBUF+130      ;LAST DATA RECORD, GET # BYTES
3254  F005  8E 8A 02    NLR:    STX     BLIM
3255  F008  4C D6 EF            JMP     GBYTE           ;GET NEXT BYTE
3256  F00B  C6 3F       ATEOF:  DEC     FEOF            ;SET FEOF
3257  F00D  A0 88       ISEOF:  LDY     #EOFERR         ;ENDFILE STATUS
3258  F00F  60                  RTS
```

```
3259                            .PAGE
3260                    ;
3261                    ; PUT BYTE TO BUFFER
3262                    ;
3263  F010  A6 3D      PBYTE:  LDX     BPTR            ;BUFFER POINTER
3264  F012  9D 00 04          STA     CASBUF+3,X      ;STORE CHAR AWAY
3265  F015  E6 3D             INC     BPTR            ;BUMP POINTER
3266  F017  A0 01             LDY     #SUCCES         ;OK STATUS
3267  F019  E0 7F             CPX     #127            ;IF BUFFER FULL
3268  F01B  F0 01             BEQ     *+3
3269  F01D  60                RTS
3270                    ; WRITE OUT THE BUFFER
3271  F01E  A9 FC             LDA     #DTA            ;RECORD TYPE = DATA
3272  F020  20 D2 F0          JSR     WSIOSB          ;DO WRITE ON SYSTEM BUFFER
3273  F023  A9 00             LDA     #0
3274  F025  85 3D             STA     BPTR            ;RESET BUFFER POINTER
3275  F027  60                RTS                     ;EXIT.
```

```
    3276                            .PAGE
    3277                        ;
    3278                        ; STATUS - RETURN STATUS INFO THRU DVSTAT
    3279                        ;
    3280  F028  A0 01          STATU:  LDY     #SUCCES
    3281  F02A  60                    RTS
```

```
3282                            .PAGE
3283                    ;
3284                    ; CLOSE
3285                    ;
3286  F02B  AD 89 02    CLOSEC: LDA     WMODE           ;SEE IF WRITING
3287  F02E  30 08               BMI     CLWRT           ;GO CLOSE FOR WRITE
3288                    ; CLOSE FOR READ - FLAG CLOSED
3289  F030  A0 01               LDY     #SUCCES         ;SUCCESSFULL
3290  F032  A9 3C       FCAX:   LDA     #MOTRST         ;STOP THE MOTOR IN CASE WAS SHORT IRG MODE
3291  F034  8D 02 D3           STA     PACTL
3292  F037  60                  RTS
3293  F038  A6 3D       CLWRT:  LDX     BPTR            ;BUFFER POINTER
3294  F03A  F0 0A               BEQ     WTLR            ;IF NO DATA BYTES IN BUFFER, NO DT1 REC
3295  F03C  8E 7F 04            STX     CASBUF+130      ;WRITE TO LAST RECORD
3296  F03F  A9 FA               LDA     #DT1            ;REC TYPE
3297  F041  20 D2 F0            JSR     WSIOSB          ;WRITE OUT USER BUFFER
3298  F044  30 EC               BMI     FCAX            ;GO IF ERROR
3299  F046  A2 7F       WTLR:   LDX     #127            ;ZERO BUFFER
3300  F048  A9 00               LDA     #0
3301  F04A  9D 00 04    ZTBUF:  STA     CASBUF+3,X
3302  F04D  CA                  DEX
3303  F04E  10 FA               BPL     ZTBUF
3304  F050  A9 FE               LDA     #EOT            ;WRITE EOT RECORD
3305  F052  20 D2 F0            JSR     WSIOSB
3306  F055  4C 32 F0            JMP     FCAX            ;FLAG CLOSED AND EXIT
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3307
    3308
    3309
    3310
    3311
    3312
    3313
    3314  F058  85 40
    3315  F05A  A5 14
    3316  F05C  18
    3317
    3318
    3319
    3320
    3321  F05D  69 1E
    3322
    3323  F05F  AA
    3324  F060  A9 FF
    3325  F062  8D 1F D0
    3326  F065  A9 00
    3327  F067  A0 F0
    3328  F069  88
    3329  F06A  D0 FD
    3330  F06C  8D 1F D0
    3331  F06F  A0 F0
    3332  F071  88
    3333  F072  D0 FD
    3334  F074  E4 14
    3335  F076  D0 E8
    3336  F078  C6 40
    3337  F07A  F0 0B
    3338  F07C  8A
    3339  F07D  18
    3340
    3341
    3342
    3343
    3344  F07E  69 0A
    3345
    3346  F080  AA
    3347  F081  E4 14
    3348  F083  D0 FC
    3349  F085  F0 D3
    3350  F087  20 8C F0
    3351  F08A  98
    3352  F08B  60
    3353  F08C  AD 25 E4
    3354  F08F  48
    3355  F090  AD 24 E4
    3356  F093  48
    3357  F094  60
    3358
    3359
    3360
```

```
        .PAGE
;
; SUBROUTINES
;
; BEEP - GENERATE TONE ON KEYBOARD SPEAKER
; ON ENTRY A= FREQ
;
BEEP:   STA     FREQ
BEEP1:  LDA     RTCLOK+2        ;CURRENT CLOCK
        CLC
        .IF     PALFLG
        ADC     #25
        .ENDIF
        .IF     PALFLG-1
        ADC     #30             ;1 SEC TONE
        .ENDIF
        TAX
WFL:    LDA     #$FF
        STA     CONSOL          ;TURN ON SPEAKER
        LDA     #0
        LDY     #$F0
        DEY
        BNE     *-1
        STA     CONSOL          ;TURN OFF SPEAKER
        LDY     #$F0
        DEY
        BNE     *-1
        CPX     RTCLOK+2        ;SEE IF 1 SEC IS UP YET
        BNE     WFL
        DEC     FREQ            ;COUNT BEEPS
        BEQ     WFAK            ;IF ALL DONE GO WAIT FOR KEY
        TXA
        CLC
        .IF     PALFLG
        ADC     #8
        .ENDIF
        .IF     PALFLG-1
        ADC     #10
        .ENDIF
        TAX
        CPX     RTCLOK+2
        BNE     *-2
        BEQ     BEEP1           ;UNCOND GO BEEP AGIN
WFAK:   JSR     WFAK1           ;USE SIMULATED "JMP (KGETCH)"
        TYA
        RTS
WFAK1:  LDA     KEYBDV+5
        PHA
        LDA     KEYBDV+4        ;SIMULATE "JMP (KGETCH)"
        PHA
        RTS
;
; SIOSB - CALL SIO ON SYSTEM BUFFER
;
```

```
ERR LINE  ADDR  B1 B2 B3 B4

    3361  F095  8D 02 03          SIOSB:  STA     DCOMND      ;SAVE COMMAND
    3362  F098  A9 00                      LDA     #0
    3363  F09A  8D 09 03                   STA     DBYTHI      ;SET BUFFER LENGTH
    3364  F09D  A9 83                      LDA     #131
    3365  F09F  8D 08 03                   STA     DBYTLO
    3366  F0A2  A9 03                      LDA     #CBUFH
    3367  F0A4  8D 05 03                   STA     DBUFHI      ;SET BUFFER ADDRESS
    3368  F0A7  A9 FD                      LDA     #CBUFL
    3369  F0A9  8D 04 03                   STA     DBUFLO
    3370  F0AC  A9 60               CSIO:   LDA     #$60        ;CASSET PSEUDO DEVICE
    3371  F0AE  8D 00 03                   STA     DDEVIC
    3372  F0B1  A9 00                      LDA     #0
    3373  F0B3  8D 01 03                   STA     DUNIT
    3374  F0B6  A9 23                      LDA     #35         ;DEVICE TIMEOUT (5/30/79)
    3375  F0B8  8D 06 03                   STA     DTIMLO
    3376  F0BB  AD 02 03                   LDA     DCOMND      ;GET COMMAND BACK
    3377  F0BE  A0 40                      LDY     #SRSTA      ;SIO READ STATUS COMMAND
    3378  F0C0  C9 52                      CMP     #'R
    3379  F0C2  F0 02                      BEQ     *+4
    3380  F0C4  A0 80                      LDY     #SWSTA      ;SIO WRITE STATUS COMMAND
    3381  F0C6  8C 03 03                   STY     DSTATS      ;SET STATUS FOR SIO
    3382  F0C9  A5 3E                      LDA     FTYPE
    3383  F0CB  8D 0B 03                   STA     DAUX2       ;INDICATE IF SHORT IRG MODE
    3384  F0CE  20 59 E4                   JSR     SIOV        ;GO CALL SIO
    3385  F0D1  60                        RTS
    3386                           ;
    3387                           ; WSIOSB - WRITE SIO SYSTEM BUFFER
    3388                           ;
    3389  F0D2  8D FF 03           WSIOSB: STA     CASBUF+2    ;STORE TYPE BYTE
    3390  F0D5  A9 55                      LDA     #$55
    3391  F0D7  8D FD 03                   STA     CASBUF+0
    3392  F0DA  8D FE 03                   STA     CASBUF+1
    3393  F0DD  A9 57                      LDA     #'W ;WRITE
    3394  F0DF  20 95 F0                   JSR     SIOSB       ;CALL SIO ON SYSTEM BUFFER
    3395  F0E2  60                        RTS     AND         ;RETURN
    3396  F0E3                     CRNTP6  =*
    3397                                   *=$14
    3398  0014  00                 CASSPR: .BYTE   MONORG-CRNTP6 ;^GCASCV IS TOO LONG
```

```
3399
3400                            .TITLE  'MONITOR  ***** MONITP.SRC ***** 3/9/79 *****  4:00:00 P
3401                    ;
3402                    ;
3403                    ;
3404                    ;       CONSTANT EQUATES
3405                    ;
3406  0009              PUTTXT  =       $9          ;"PUT TEXT RECORD" CIO COMMAND CODE
3407  0007              GETCAR  =       $7          ;"GET CHARACTER" CIO COMMAND CODE
3408  000B              PUTCAR  =       $B          ;"PUT CHARACTER" CIO COMMAND CODE
3409  0000              INIMLL  =       $00         ;INITIAL MEM LO LOW BYTE
3410  0007              INIMLH  =       $07         ;INITIAL MEM LO HIGH BYTE
3411                    ; GOOD  =       $1          ;GOOD STATUS CODE
3412                    ; WRITE =       $57         ;WRITE COMMAND
3413                    ; READ  =       $52         ;READ COMMAND
3414                    ; STATC =       $53         ;STATUS COMMAND
3415  0000              SEX     =       $0          ;SCREEN EDITOR IOCB INDEX
3416  007D              CLS     =       $7D         ;CLEAR SCREEN CODE
3417  0092              CTRLC   =       $92         ;KEYBOARD CODE FOR 'CONTROL C'
3418  0088              EOF     =       $88         ;CASSETTE END OF FILE CODE
3419  0000              LIRG    =       $0          ;LONG IRG TYPE CODE
3420                    ;
3421  0004              BUFFH   =       (CASBUF+3)/256
3422  0000              BUFFL   =       (-256)*BUFFH+CASBUF+3 ;BUFFER POINTER
3423                    ;
3424                    ;
3425                    ;
3426                    ; THE FOLLOWING EQUATES ARE IN THE CARTRIDGE ADDRESS SPACE.
3427                    ;
3428                    ;
3429                    ; "B" CARTRIDGE ADDR'S ARE 8000-9FFF (36K CONFIG. ONLY)
3430                    ; "A" CART. ADDR'S ARE A000-BFFF (36K CONFIG. ONLY)
3431                    ;
3432                    ; "A" CART. ADDR'S ARE B000-BFFF (48K CONFIG. ONLY)
3433                    ;
3434                            *=$BFFA
3435  BFFA              CARTCS: .RES    2           ;CARTRIDGE COLD START ADDRESS.
3436  BFFC              CART:   .RES    1           ;CARTRIDGE AVAILABLE FLAG BYTE.
3437  BFFD              CARTFG: .RES    1           ;CARTRIDGE FLAG BYTE.  BIT 0=FLAG1,
3438  BFFE              CARTAD: .RES    2           ;2-BYTE CARTRIDGE START VECTOR
3439                    ;
3440                    ;
3441                    ;       CARTRIDGE FLAG ACTION DEFINITIONS
3442                    ;
3443                    ;
3444                    ;       BIT             ACTION IF SET
3445                    ;
3446                    ;       7               SPECIAL -- DON'T POWER-UP, JUST RUN CARTRIDGE
3447                    ;       6-3             NONE
3448                    ;       2               RUN CARTRIDGE
3449                    ;       1               NONE
3450                    ;       0               BOOT DOS
3451                    ;
3452                    ;
```

```
3453                        ;       *****
3454                        ;       NOTE
3455                        ;       *****
3456                        ;
3457                        ;       1. IF BIT2 IS 0, GOTO BLACKBOARD MODE.
3458                        ;       2. IF BIT0 SET, THE DISK WILL BE BOOTED BEFORE ANY
3459                        ;          OTHER ACTION.
3460                        ;
3461                        ;
3462                        ;
3463                        ;
3464                        ;
3465                        ;
3466                        ;
3467                        ;
3468                        ;
3469                        ;       POWER-UP VECTOR
3470                        ;
3471                        ;*****************************
3472                        ;       *=$FFFC
3473                        ;
3474                        ; PVECT .WORD   PWRUP           POWER-UP VECTOR
3475                        ;*************************
3476                        ;
3477                        ;
3478                        ;
3479                        ;
3480                        ;
3481                        ;       ENTRY POINT VECTOR
3482                        ;
3483                                *=BLKBDV
3484                        ;
3485  E471  4C 23 F2                JMP     SIGNON          ;BLACK BOARD VECTOR
3486                        ;
3487                                *=WARMSV
3488                        ;
3489  E474  4C 1B F1                JMP     RESET           ;WARM START VECTOR
3490                        ;
3491                                *=COLDSV
3492                        ;
3493  E477  4C 25 F1                JMP     PWRUP           ;COLD START VECTOR (9000 FOR RAM VECTOR WRIT
3494                        ;
3495                                *=$9000
3496  9000  20 0C 90                JSR     $900C
3497  9003  4C 25 F1                JMP     PWRUP           ;(TO HANDLE RAM VECTOR WRITING)
3498  9006  20 0C 90                JSR     $900C
3499  9009  4C 1B F1                JMP     RESET
3500                        ;
3501                        ;
3502                        ;
3503                                *=MONORG
3504                        ;
3505                        ;
3506                        ;
```

```
3507                                ;
3508                                ;       HANDLER TABLE ENTRIES
3509                                ;
3510  F0E3  50                      TBLENT:  .BYTE   'P'
3511  F0E4  30 E4                            .WORD   PRINTV
3512  F0E6  43                               .BYTE   'C'
3513  F0E7  40 E4                            .WORD   CASETV
3514  F0E9  45                               .BYTE   'E'
3515  F0EA  00 E4                            .WORD   EDITRV
3516  F0EC  53                               .BYTE   'S'
3517  F0ED  10 E4                            .WORD   SCRENV
3518  F0EF  4B                               .BYTE   'K'
3519  F0F0  20 E4                            .WORD   KEYBDV
3520                                ;
3521                                ;
3522                                ;TBLLEN  =       IDENT-TBLENT-1  HANDLER TABLE LENGTH.   "MOVED TO LINE 8
3523                                ;
3524                                ;       ***** PRINT MESSAGES *****
3525                                ;
3526                                ;
3527  F0F2  7D 41 54 41             IDENT:  .BYTE   CLS,'ATARI COMPUTER - MEMO PAD',CR
3528  F0F6  52 49 20 43
3529  F0FA  4F 4D 50 55
3530  F0FE  54 45 52 20
3531  F102  2D 20 4D 45
3532  F106  4D 4F 20 50
3533  F10A  41 44 9B
3534                                ;
3535  00F0                          IDENTH  =       IDENT/256
3536  00F2                          IDENTL  =       (-256)*IDENTH+IDENT ;SYSTEM I.D. MSG POINTER
3537                                ;
3538  000E                          TBLLEN  =       IDENT-TBLENT-1 ;HANDLER TABLE LENGTH
3539  F10D  42 4F 4F 54             DERR5:  .BYTE   'BOOT ERROR',CR
3540  F111  20 45 52 52
3541  F115  4F 52 9B
3542                                ;
3543  00F1                          DERRH   =       DERR5/256
3544  000D                          DERRL   =       (-256)*DERRH+DERR5 ;DISK ERROR MSG POINTER
3545                                ;
3546                                ;
3547                                ;
3548                                ;
3549                                ;       DEVICE/FILENAME SPECIFICATIONS
3550                                ;
3551  F118  45 3A 9B                OPNEDT: .BYTE   'E:',CR ;"OPEN SCREEN EDITOR" DEVICE SPEC.
3552                                ;
3553  00F1                          OPNH    =       OPNEDT/256
3554  0018                          OPNL    =(-256)*OPNH+OPNEDT ;SCREEN EDITOR OPEN POINTER
3555  F11B                                
3556                                ;
3557                                ;
3558                                ;
3559                                ;
3560                                ;*****************************************************************
```

```
3561                            ;       RESET BUTTON ROUTINE STARTS HERE
3562                            ;*************************************************************
3563                            ;
3564  F11B  78                 RESET:  SEI                     ;DISABLE IRQ INTERRUPTS
3565  F11C  AD 44 02                    LDA     COLDST        ;WERE WE IN MIDDLE OF COLDSTART?
3566  F11F  D0 04                       BNE     PWRUP         ;YES, GO TRY IT AGAIN
3567  F121  A9 FF                       LDA     #$FF
3568  F123  D0 03                       BNE     PWRUP1        ;SET WARM START FLAG
3569                            ;
3570                            ;
3571                            ;
3572                            ;*************************************************************
3573                            ;       POWER UP ROUTINES START HERE
3574                            ;*************************************************************
3575                            ;
3576  F125  78                 PWRUP:  SEI                     ;DISABLE IRQ INTERRUPTS
3577  F126  A9 00                       LDA     #0            ;CLEAR WARMSTART FLAG
3578  F128  85 08              PWRUP1: STA     WARMST
3579  F12A  D8                          CLD                   ;CLEAR DECIMAL FLAG.
3580  F12B  A2 FF                       LDX     #$FF
3581  F12D  9A                          TXS                   ;SET STACK POINTER
3582  F12E  20 3F F2                    JSR     SPECL         ;CARTRIDGE SPECIAL CASE?
3583  F131  20 77 F2                    JSR     HARDI         ;DO HARDWARE INITIALIZATION
3584  F134  A5 08                       LDA     WARMST        ;IS IT WARMSTART?
3585  F136  D0 28                       BNE     ZOSRAM        ;YES, ONLY ZERO OS RAM
3586                            ;
3587  F138  A9 00              ZERORM: LDA     #0
3588  F13A  A0 08                       LDY     #WARMST
3589  F13C  85 04                       STA     RAMLO
3590  F13E  85 05                       STA     RAMLO+1       ;INITIALIZE RAM POINTER
3591  F140  91 04              CLRRAM: STA     (RAMLO),Y     ;CLEAR MEMORY LOC.
3592  F142  C8                          INY
3593  F143  C0 00                       CPY     #0            ;AT END OF PAGE?
3594  F145  D0 F9                       BNE     CLRRAM
3595  F147  E6 05                       INC     RAMLO+1       ;YES, INCR PAGE POINTER
3596  F149  A6 05                       LDX     RAMLO+1
3597  F14B  E4 06                       CPX     TRAMSZ        ;AT END OF MEM?
3598  F14D  D0 F1                       BNE     CLRRAM        ;NO.
3599                            ;
3600                            ;  INITIALIZE DOSVEC TO POINT TO SIGNON (BLACKBOARD)
3601  F14F  AD 72 E4                    LDA     BLKBDV+1
3602  F152  85 0A                       STA     DOSVEC        ;USE BLACKBOARD VECTOR
3603  F154  AD 73 E4                    LDA     BLKBDV+2      ;FOR DOSVEC
3604  F157  85 0B                       STA     DOSVEC+1
3605  F159  A9 FF                       LDA     #$FF
3606  F15B  8D 44 02                    STA     COLDST        ;SET TO SHOW IN MIDDLE OF COLDSTART
3607  F15E  D0 13                       BNE     ESTSCM        ;GO AROUND ZOSRAM
3608                            ;
3609                            ;  CLEAR OS RAM (FOR WARMSTART)
3610  F160  A2 00              ZOSRAM: LDX     #0
3611  F162  8A                          TXA
3612  F163  9D 00 02           ZOSRM2: STA     $200,X        ;CLEAR PAGES 2 AND 3
3613  F166  9D 00 03                    STA     $300,X
3614  F169  CA                          DEX
```

```
3615  F16A  D0 F7          BNE     ZOSRM2
3616  F16C  A2 10          LDX     #INTZBS
3617  F16E  95 00  ZOSRM3: STA     0,X          ;CLEAR ZERO PAGE LOCATIONS INTZBS-7F
3618  F170  E8             INX
3619  F171  10 FB          BPL     ZOSRM3
3620                       ;
3621                       ; ESTABLISH SCREEN MARGINS
3622  F173  A9 02  ESTSCM: LDA     #LEDGE
3623  F175  85 52          STA     LMARGN
3624  F177  A9 27          LDA     #REDGE
3625  F179  85 53          STA     RMARGN
3626                       ;
3627                       ;
3628                       ; MOVE VECTOR TABLE FROM ROM TO RAM
3629  F17B  A2 25  OPSYS:  LDX     #$25
3630  F17D  BD 80 E4 MOVVEC: LDA    VCTABL,X     ;ROM TABLE
3631  F180  9D 00 02       STA     INTABS,X     ;TO RAM
3632  F183  CA             DEX
3633  F184  10 F7          BPL     MOVVEC
3634  F186  20 8A F2       JSR     OSRAM        ;DO O.S. RAM SETUP
3635  F189  58             CLI                  ;ENABLE IRQ INTERRUPTS
3636                       ;
3637                       ;
3638                       ;       LINK HANDLERS
3639                       ;
3640  F18A  A2 0E          LDX     #TBLLEN
3641  F18C  BD E3 F0 NXTENT: LDA    TBLENT,X     ;READ HANDLER TABLE ENTRY
3642  F18F  9D 1A 03       STA     HATABS,X     ;PUT IN TABLE
3643  F192  CA             DEX
3644  F193  10 F7          BPL     NXTENT       ;DONE WITH ALL ENTRIES?
3645                       ;
3646                       ;
3647                       ;
3648                       ;
3649                       ;
3650                       ; INTERROGATE CARTRIDGE ADDR. SPACE TO SEE WHICH CARTRIDGES THERE ARE
3651                       ;
3652  F195  A2 00          LDX     #0
3653  F197  86 07          STX     TSTDAT       ;CLEAR "B" CART. FLAG
3654  F199  86 06          STX     TRAMSZ       ;CLEAR "A" CART. FLAG
3655  F19B  AE E4 02       LDX     RAMSIZ
3656  F19E  E0 90          CPX     #$90         ;RAM IN "B" CART. SLOT?
3657  F1A0  B0 0A          BCS     ENDBCK
3658  F1A2  AD FC 9F       LDA     CART-$2000   ;NO,
3659  F1A5  D0 05          BNE     ENDBCK       ;CART. PLUGGED INTO "B" SLOT?
3660  F1A7  E6 07          INC     TSTDAT       ;YES, SET "B" CART. FLAG
3661  F1A9  20 3C F2       JSR     CBINI        ;INITIALIZE CARTRIDGE "B"
3662                       ;
3663  F1AC  AE E4 02 ENDBCK: LDX    RAMSIZ
3664  F1AF  E0 B0          CPX     #$B0         ;RAM IN "A" CART. SLOT?
3665  F1B1  B0 0A          BCS     ENDACK
3666  F1B3  AE FC BF       LDX     CART         ;NO,
3667  F1B6  D0 05          BNE     ENDACK       ;CART. PLUGGED INTO "A" SLOT?
3668  F1B8  E6 06          INC     TRAMSZ       ;YES, SET "A" CART. FLAG
```

```
3669  F1BA  20 39 F2          JSR     CAINI       ;INITIALIZE CARTRIDGE "A"
3670                          ;
3671                          ;
3672                          ; OPEN SCREEN EDITOR
3673                          ;
3674  F1BD  A9 03             ENDACK: LDA     #3
3675  F1BF  A2 00                     LDX     #SEX
3676  F1C1  9D 42 03                  STA     ICCOM,X     ;OPEN I/O COMMAND
3677  F1C4  A9 18                     LDA     #OPNL
3678  F1C6  9D 44 03                  STA     ICBAL,X
3679  F1C9  A9 F1                     LDA     #OPNH
3680  F1CB  9D 45 03                  STA     ICBAH,X     ;SET BUFFER POINTER TO OPEN SCREEN EDITOR
3681  F1CE  A9 0C                     LDA     #$C
3682  F1D0  9D 4A 03                  STA     ICAX1,X     ;SET UP OPEN FOR INPUT/OUTPUT
3683  F1D3  20 56 E4                  JSR     CIOV        ;GO TO CIO
3684                          ;
3685  F1D6  10 03                     BPL     SCRNOK      ;BR IF NO ERROR
3686  F1D8  4C 25 F1                  JMP     PWRUP       ;RETRY PWRUP IF ERROR (SHOULD NEVER HAPPEN!)
3687  F1DB  E8                SCRNOK: INX                 ;SCREEN OK, SO WAIT FOR VBLANK TO
3688  F1DC  D0 FD                     BNE     SCRNOK      ;BRING UP THE DISPLAY
3689  F1DE  C8                        INY
3690  F1DF  10 FA                     BPL     SCRNOK
3691                          ;
3692                          ;
3693                          ; DO CASSETTE BOOT
3694  F1E1  20 B2 F3                  JSR     CSBOOT      ;CHECK, BOOT, AND INIT
3695                          ;
3696                          ; CHECK TO SEE IF EITHER CARTRIDGE WANTS DISK BOOT
3697  F1E4  A5 06                     LDA     TRAMSZ      ;CHECK BOTH CARTRIDGES
3698  F1E6  05 07                     ORA     TSTDAT      ;
3699  F1E8  F0 12                     BEQ     NOCART      ;NEITHER CARTRIDGE LIVES
3700  F1EA  A5 06                     LDA     TRAMSZ      ;"A" CART?
3701  F1EC  F0 03                     BEQ     NOA1        ;NO
3702  F1EE  AD FD BF                  LDA     CARTFG      ;GET CARTRIDGE MODE FLAG
3703  F1F1  A6 07             NOA1:   LDX     TSTDAT      ;"B" CART?
3704  F1F3  F0 03                     BEQ     NOB1        ;NO
3705  F1F5  0D FD 9F                  ORA     CARTFG-$2000 ;ADD OTHER FLAG
3706  F1F8  29 01             NOB1:   AND     #1          ;DOES EITHER CART WANT BOOT?
3707  F1FA  F0 03                     BEQ     NOBOOT      ;NO
3708                          ;
3709                          ; DO DISK BOOT
3710  F1FC  20 CF F2          NOCART: JSR     BOOT        ;CHECK, BOOT, AND INIT
3711                          ;
3712                          ; GO TO ONE OF THE CARTRIDGES IF THEY SO DESIRE
3713  F1FF  A9 00             NOBOOT: LDA     #0
3714  F201  8D 44 02                  STA     COLDST      ;RESET TO SHOW DONE WITH COLDSTART
3715  F204  A5 06                     LDA     TRAMSZ      ;"A" CART?
3716  F206  F0 0A                     BEQ     NOA2        ;NO
3717  F208  AD FD BF                  LDA     CARTFG      ;GET CARTRIDGE MODE FLAG
3718  F20B  29 04                     AND     #4          ;DOES IT WANT TO RUN?
3719  F20D  F0 03                     BEQ     NOA2        ;NO
3720  F20F  6C FA BF                  JMP     (CARTCS)    ;RUN "A" CARTRIDGE
3721  F212  A5 07             NOA2:   LDA     TSTDAT      ;"B" CART?
3722  F214  F0 0A                     BEQ     NOCAR2      ;NO
```

```
3723  F216  AD FD 9F          LDA     CARTFG-$2000 ;GET "B" MODE FLAG
3724  F219  29 04             AND     #4           ;DOES IT WANT TO RUN?
3725  F21B  F0 DF             BEQ     NOCART       ;NO
3726  F21D  6C FA 9F          JMP     (CARTCS-$2000) ;RUN "B" CARTRIDGE
3727                        ;
3728                        ; NO CARTRIDGES, OR NEITHER WANTS TO RUN,
3729                        ; SO GO TO DOSVEC (DOS, CASSETTE, OR BLACKBOARD)
3730  F220  6C 0A 00       NOCAR2: JMP     (DOSVEC)
3731                        ;
3732                        ; PRINT SIGN-ON MESSAGE
3733  F223  A2 F2           SIGNON: LDX     #IDENTL
3734  F225  A0 F0                   LDY     #IDENTH
3735  F227  20 85 F3                JSR     PUTLIN       ;GO PUT SIGN-ON MSG ON SCREEN
3736                        ;
3737                        ;
3738                        ;       BLACKBOARD ROUTINE
3739
3740  F22A  20 30 F2        BLACKB: JSR     BLKB2        ;"JSR EGETCH"
3741  F22D  4C 2A F2                JMP     BLACKB       ;FOREVER
3742  F230  AD 05 E4        BLKB2:  LDA     EDITRV+5     ;HIGH BYTE
3743  F233  48                     PHA
3744  F234  AD 04 E4                LDA     EDITRV+4     ;LOW BYTE
3745  F237  48                     PHA
3746  F238  60                     RTS                  ;SIMULATES "JMP (EDITRV)"
3747                        ;
3748                        ;
3749                        ; CARTRIDGE INITIALIZATION INDIRECT JUMPS
3750  F239  6C FE BF        CAINI:  JMP     (CARTAD)
3751  F23C  6C FE 9F        CBINI:  JMP     (CARTAD-$2000)
```

```
3752                                    .PAGE
3753                                ;
3754                                ;
3755                                ;
3756                                ;
3757                                ;
3758                                ;
3759                                ;               S U B R O U T I N E S
3760                                ;
3761                                ;
3762                                ;
3763                                ;
3764                                ;
3765                                ;
3766                                ;
3767                                ;
3768                                ;
3769                                ;
3770                                ;
3771                                ;
3772                                ;
3773                                ;
3774                                ;
3775                                ;
3776                                ;
3777                                ;
3778                                ; CHECK FOR HOW MUCH RAM & SPECIAL CARTRIDGE CASE.
3779                                ; IF SPECIAL CARTRIDGE CASE, DON'T GO BACK -- GO TO CART.
3780                                ;
3781  F23F  AD FC BF                SPECL:  LDA     CART            ;CHECK FOR RAM OR CART
3782  F242  D0 13                           BNE     ENSPE2          ;GO IF NOTHING OR MAYBE RAM
3783  F244  EE FC BF                        INC     CART            ;NOW DO RAM CHECK
3784  F247  AD FC BF                        LDA     CART            ;IS IT ROM?
3785  F24A  D0 08                           BNE     ENSPEC          ;NO
3786  F24C  AD FD BF                        LDA     CARTFG          ;YES,
3787  F24F  10 03                           BPL     ENSPEC          ;BIT SET?
3788  F251  6C FE BF                        JMP     (CARTAD)        ;YES, GO RUN CARTRIDGE
3789                                ;
3790                                ; CHECK FOR AMOUNT OF RAM
3791                                ;
3792                                ;
3793  F254  CE FC BF                ENSPEC: DEC     CART            ;RESTORE RAM IF NEEDED
3794  F257  A0 00                   ENSPE2: LDY     #0
3795  F259  84 05                           STY     RAMLO+1
3796  F25B  A9 10                           LDA     #$10
3797  F25D  85 06                           STA     TRAMSZ          ;SET RAM POINTER TO 4K.
3798  F25F  B1 05                   HOWMCH: LDA     (RAMLO+1),Y     ;READ RAM LOCATION
3799  F261  49 FF                           EOR     #$FF            ;INVERT IT.
3800  F263  91 05                           STA     (RAMLO+1),Y     ;WRITE INVERTED DATA.
3801  F265  D1 05                           CMP     (RAMLO+1),Y     ;READ RAM AGAIN
3802  F267  D0 0D                           BNE     ENDRAM
3803  F269  49 FF                           EOR     #$FF            ;CONVERT IT BACK
3804  F26B  91 05                           STA     (RAMLO+1),Y     ;RESTORE ORIGINAL RAM DATA
3805  F26D  A5 06                           LDA     TRAMSZ
```

```
3806  F26F  18                  CLC
3807  F270  69 10               ADC     #$10
3808  F272  85 06               STA     TRAMSZ      ;INCR. RAM POINTER BY 4K.
3809  F274  D0 E9               BNE     HOWMCH      ;GO FIND HOW MUCH RAM.
3810  F276  60          ENDRAM:  RTS
3811                    ;
3812                    ;
3813                    ;
3814                    ;
3815                    ;       HARDWARE INITIALIZATION
3816                    ;
3817                    ;
3818  F277  A9 00       HARDI:   LDA     #0
3819  F279  AA                   TAX
3820  F27A  9D 00 D0    CLRCHP:  STA     $D000,X
3821  F27D  9D 00 D4             STA     $D400,X
3822  F280  9D 00 D2             STA     $D200,X
3823  F283  9D 00 D3             STA     $D300,X
3824  F286  E8                   INX
3825  F287  D0 F1                BNE     CLRCHP
3826  F289  60                   RTS
3827
3828
3829                    ;       O.S.  RAM SETUP
3830                    ;
3831  F28A  C6 11       OSRAM:   DEC     BRKKEY      ;TURN OFF BREAK KEY FLAG
3832  F28C  A9 54                LDA     #.LOW.BRKKY2
3833  F28E  8D 36 02             STA     BRKKY
3834  F291  A9 E7                LDA     #.HIGH.BRKKY2
3835  F293  8D 37 02             STA     BRKKY+1
3836  F296  A5 06                LDA     TRAMSZ      ;READ RAM SIZE IN TEMP. REG.
3837  F298  8D E4 02             STA     RAMSIZ      ;SAVE IT IN RAM SIZE.
3838  F29B  8D E6 02             STA     MEMTOP+1    ; INIT. MEMTOP ADDR HI BYTE
3839  F29E  A9 00                LDA     #0
3840  F2A0  8D E5 02             STA     MEMTOP      ;INIT. MEMTOP ADDR LO BYTE
3841  F2A3  A9 00                LDA     #INIMLL
3842  F2A5  8D E7 02             STA     MEMLO
3843  F2A8  A9 07                LDA     #INIMLH
3844  F2AA  8D E8 02             STA     MEMLO+1     ;INITIALIZE MEMLO ADDR VECTOR
3845  F2AD  20 0C E4             JSR     EDITRV+$C   ;EDITOR INIT.
3846  F2B0  20 1C E4             JSR     SCRENV+$C   ;SCREEN INIT.
3847  F2B3  20 2C E4             JSR     KEYBDV+$C   ;KEYBOARD INIT.
3848  F2B6  20 3C E4             JSR     PRINTV+$C   ;PRINTER HANDLER INIT
3849  F2B9  20 4C E4             JSR     CASETV+$C   ;CASSETTE HANDLER INIT
3850  F2BC  20 6E E4             JSR     CIOINV      ;CIO INIT.
3851  F2BF  20 65 E4             JSR     SIOINV      ;SIO INIT.
3852  F2C2  20 6B E4             JSR     INTINV      ;INTERRUPT HANDLER INIT.
3853  F2C5  AD 1F D0             LDA     CONSOL
3854  F2C8  29 01                AND     #$1
3855  F2CA  D0 02                BNE     NOKEY       ;GAME START KEY DEPRESSED?
3856  F2CC  E6 4A                INC     CKEY        ;YES, SET KEY FLAG.
3857  F2CE  60          NOKEY:   RTS
3858                    ;
3859                    ;
```

```
3860                    ; DO BOOT OF DISK
3861                    ;
3862 F2CF A5 08         BOOT:   LDA     WARMST
3863 F2D1 F0 0A                 BEQ     NOWARM          ;WARM START?
3864 F2D3 A5 09                 LDA     BOOT?           ;YES,
3865 F2D5 29 01                 AND     #1
3866 F2D7 F0 03                 BEQ     NOINIT          ;VALID BOOT?
3867 F2D9 20 7E F3              JSR     DINI            ;YES, RE-INIT. DOS SOFTWARE
3868 F2DC 60            NOINIT: RTS
3869 F2DD A9 01         NOWARM: LDA     #1
3870 F2DF 8D 01 03              STA     DUNIT           ;ASSIGN DISK DRIVE NO.
3871 F2E2 A9 53                 LDA     #STATC
3872 F2E4 8D 02 03              STA     DCOMND          ;SET UP STATUS COMMAND
3873 F2E7 20 53 E4              JSR     DSKINV          ;GO DO DISK STATUS
3874 F2EA 10 01                 BPL     DOBOOT          ;IS STATUS FROM SIO GOOD?
3875 F2EC 60                    RTS                     ;NO, GO BACK WITH BAD BOOT STATUS
3876                    ;
3877 F2ED A9 00         DOBOOT: LDA     #0
3878 F2EF 8D 0B 03              STA     DAUX2
3879 F2F2 A9 01                 LDA     #1
3880 F2F4 8D 0A 03              STA     DAUX1           ;SET SECTOR # TO 1.
3881 F2F7 A9 00                 LDA     #BUFFL
3882 F2F9 8D 04 03              STA     DBUFLO
3883 F2FC A9 04                 LDA     #BUFFH
3884 F2FE 8D 05 03              STA     DBUFHI          ;SET UP BUFFER ADDR
3885 F301 20 9D F3      SECT1:  JSR     GETSEC          ;GET SECTOR
3886 F304 10 08                 BPL     ALLSEC          ;STATUS O.K. ?
3887 F306 20 81 F3      BADDSK: JSR     DSKRDE          ;NO, GO PRINT DISK READ ERROR
3888 F309 A5 4B                 LDA     CASSBT
3889 F30B F0 E0                 BEQ     DOBOOT          ;CASSETTE BOOT?
3890 F30D 60                    RTS                     ;YES, QUIT
3891 F30E A2 03         ALLSEC: LDX     #3
3892 F310 BD 00 04      RDBYTE: LDA     CASBUF+3,X      ;READ A BUFFER BYTE
3893 F313 9D 40 02              STA     DFLAGS,X        ;STORE IT
3894 F316 CA                    DEX
3895 F317 10 F7                 BPL     RDBYTE          ;DONE WITH 4 BYTE TRANSFER ?
3896 F319 AD 42 02              LDA     BOOTAD          ;YES,
3897 F31C 85 04                 STA     RAMLO
3898 F31E AD 43 02              LDA     BOOTAD+1
3899 F321 85 05                 STA     RAMLO+1         ;PUT BOOT ADDR INTO Z. PAGE RAM
3900 F323 AD 04 04              LDA     CASBUF+7
3901 F326 85 0C                 STA     DOSINI          ;ESTABLISH DOS INIT ADDRESS
3902 F328 AD 05 04              LDA     CASBUF+8
3903 F32B 85 0D                 STA     DOSINI+1
3904 F32D A0 7F         MVBUFF: LDY     #$7F            ;YES, SET BYTE COUNT
3905 F32F B9 00 04      MVNXB:  LDA     CASBUF+3,Y
3906 F332 91 04                 STA     (RAMLO),Y       ;MOVE A BYTE FROM SECTOR BUFFER TO BOOT ADDR
3907 F334 88                    DEY
3908 F335 10 F8                 BPL     MVNXB           ;DONE ?
3909 F337 18                    CLC                     ;YES,
3910 F338 A5 04                 LDA     RAMLO
3911 F33A 69 80                 ADC     #$80
3912 F33C 85 04                 STA     RAMLO
3913 F33E A5 05                 LDA     RAMLO+1
```

```
3914 F340 69 00          ADC   #0
3915 F342 85 05          STA   RAMLO+1      ;INCR BOOT LOADER BUFFER POINTER.
3916 F344 CE 41 02       DEC   DBSECT       ;DECR # OF SECTORS.
3917 F347 F0 11          BEQ   ENBOOT       ;MORE SECTORS ?
3918 F349 EE 0A 03       INC   DAUX1        ;YES, INCR SECTOR #
3919 F34C 20 9D F3 SECTX: JSR  GETSEC       ;GO GET SECTOR.
3920 F34F 10 DC          BPL   MVBUFF       ;STATUS O.K. ?
3921 F351 20 81 F3       JSR   DSKRDE       ;NO, GO PRINT DISK READ ERROR
3922 F354 A5 4B          LDA   CASSBT
3923 F356 D0 AE          BNE   BADDSK       ;IF CASSETTE, QUIT.
3924 F358 F0 F2          BEQ   SECTX        ;IF DISK, TRY SECTOR AGAIN.
3925 F35A A5 4B ENBOOT:  LDA   CASSBT
3926 F35C F0 03          BEQ   XBOOT        ;CASSETTE BOOT ?
3927 F35E 20 9D F3       JSR   GETSEC       ;YES, GET EOF RECORD, BUT DON'T USE IT.
3928 F361 20 6C F3 XBOOT: JSR  BLOAD        ;GO EXECUTE BOOT LOADER
3929 F364 B0 A0          BCS   BADDSK       ;IF BAD BOOT, DO IT OVER AGAIN
3930 F366 20 7E F3       JSR   DINI         ;GO INIT. SOFTWARE
3931 F369 E6 09          INC   BOOT?        ;SHOW BOOT SUCCESS
3932 F36B 60             RTS
3933 F36C 18 BLOAD:  CLC
3934 F36D AD 42 02       LDA   BOOTAD
3935 F370 69 06          ADC   #6
3936 F372 85 04          STA   RAMLO
3937 F374 AD 43 02       LDA   BOOTAD+1
3938 F377 69 00          ADC   #0
3939 F379 85 05          STA   RAMLO+1      ;PUT START ADDR OF BOOTLOADER INTO RAM
3940 F37B 6C 04 00       JMP   (RAMLO)
3941 F37E 6C 0C 00 DINI: JMP   (DOSINI)
3942                     ;
3943                     ;
3944                     ;
3945                     ;
3946                     ; DISPLAY DISK READ ERROR MSG
3947                     ;
3948 F381 A2 0D DSKRDE: LDX   #DERRL
3949 F383 A0 F1         LDY   #DERRH
3950                     ;
3951                     ;
3952                     ;
3953                     ; PUT LINE ON SCREEN AT PRESENT CURSOR POSITION
3954                     ;
3955                     ;   X-REG -- LO BYTE, BEGIN ADDR OF LINE
3956                     ;   Y-REG -- HI BYTE, BEGIN ADDR OF LINE
3957                     ;
3958 F385 8A PUTLIN: TXA
3959 F386 A2 00         LDX   #SEX
3960 F388 9D 44 03      STA   ICBAL,X
3961 F38B 98            TYA
3962 F38C 9D 45 03      STA   ICBAH,X      ;SET UP ADDR OF BEGIN OF LINE
3963 F38F A9 09         LDA   #PUTTXT
3964 F391 9D 42 03      STA   ICCOM,X      ;"PUT TEXT RECORD" COMMAND
3965 F394 A9 FF         LDA   #$FF
3966 F396 9D 48 03      STA   ICBLL,X      ;SET BUFFER LENGTH
3967 F399 20 56 E4      JSR   CIOV         ;PUT LINE ON SCREEN
```

```
3968  F39C  60                RTS
3969
3970                          ;
3971                          ;
3972                          ;
3973                          ; GET SECTOR FROM DISK 0
3974                          ;
3975  F39D  A5 4B             GETSEC: LDA     CASSBT
3976  F39F  F0 03                     BEQ     DISKM           ;CASSETTE BOOT ?
3977  F3A1  4C 7A E4                  JMP     RBLOKV          ;YES, GO TO READ BLOCK ROUTINE
3978  F3A4  A9 52             DISKM:  LDA     #READ
3979  F3A6  8D 02 03                  STA     DCOMND          ;SET READ SECTOR COMMAND
3980  F3A9  A9 01                     LDA     #1
3981  F3AB  8D 01 03                  STA     DUNIT           ;SET DRIVE NO. TO DRIVE 0
3982  F3AE  20 53 E4                  JSR     DSKINV          ;GET SECTOR
3983  F3B1  60                        RTS
3984                          ;
3985                          ;
3986                          ;
3987                          ; DO CHECK FOR CASSETTE BOOT & IF SO, DO BOOT
3988                          ;
3989  F3B2  A5 08             CSBOOT: LDA     WARMST          ;WARMSTART?
3990  F3B4  F0 0A                     BEQ     CSBOT2          ;NO
3991  F3B6  A5 09                     LDA     BOOT?           ;GET BOOT FLAG
3992  F3B8  29 02                     AND     #2              ;WAS CASSETTE BOOT SUCCESFULL?
3993  F3BA  F0 03                     BEQ     NOCSB2          ;NO
3994  F3BC  20 E1 F3                  JSR     CINI            ;YES, INIT CASSETTE SOFTWARE
3995  F3BF  60                NOCSB2: RTS
3996                          ;
3997  F3C0  A5 4A             CSBOT2: LDA     CKEY
3998  F3C2  F0 1C                     BEQ     NOCSBT          ;"C" KEY FLAG SET ?
3999  F3C4  A9 80                     LDA     #$80            ;YES,
4000  F3C6  85 3E                     STA     FTYPE           ;SET LONG IRG TYPE
4001  F3C8  E6 4B                     INC     CASSBT          ;SET CASSETTE BOOT FLAG
4002  F3CA  20 7D E4                  JSR     CSOPIV          ;OPEN CASSETTE FOR INPUT
4003  F3CD  20 01 F3                  JSR     SECT1           ;DO BOOT & INIT.
4004  F3D0  A9 00                     LDA     #0
4005  F3D2  85 4B                     STA     CASSBT          ;RESET CASSETTE BOOT FLAG
4006  F3D4  85 4A                     STA     CKEY            ;CLEAR KEY FLAG
4007  F3D6  06 09                     ASL     BOOT?           ;SHIFT BOOT FLAG (NOW=2 IF SUCCESS)
4008  F3D8  A5 0C                     LDA     DOSINI
4009  F3DA  85 02                     STA     CASINI          ;MOVE INIT ADDRESS FOR CASSETTE
4010  F3DC  A5 0D                     LDA     DOSINI+1
4011  F3DE  85 03                     STA     CASINI+1
4012  F3E0  60                NOCSBT: RTS
4013                          ;
4014  F3E1  6C 02 00          CINI:   JMP     (CASINI)        ;INIT CASSETTE
4015                          ;*******************************************************************
4016                          ;
4017                          ;
4018                          ; SPARE BYTE OR MODULE TOO LONG FLAG
4019                          ;
4020  F3E4                    CRNTP7  =*
4021                          ;
```

```
4022                            *=$14
4023  0014  00                 MONSPR: .BYTE   KBDORG-CRNTP7 ;^GMONITP TOO LONG
4024                            ;
```

```
4025                            .PAGE
4026                            .TITLE  'DISPLAY HANDLER  -- 10-30-78 --  DISPLC'
4027                    ;
4028                    ; HANDLER DEPENDENT EQUATES
4029                    ;
4030  007D              CLRCOD  =       $7D             ;CLEAR SCREEN ATASCI CODE
4031  009F              CNTL1   =       $9F             ;POKEY KEY CODE FOR ^1
4032                    ;
4033  0068              FRMADR  =       SAVADR
4034  0066              TOADR   =       MLTTMP
4035                    ;
```

```
 4036                                        .PAGE
 4037
 4038
 4039                                        *=EDITRV
 4040                                    ;
 4041                                    ; SCREEN EDITOR HANDLER ENTRY POINT
 4042                                    ;
 4043  E400  FB F3                       EDITOR: .WORD   EOPEN-1
 4044  E402  33 F6                               .WORD   RETUR1-1     ;(CLOSE)
 4045  E404  3D F6                               .WORD   EGETCH-1
 4046  E406  A3 F6                               .WORD   EOUTCH-1
 4047  E408  33 F6                               .WORD   RETUR1-1     ;(STATUS)
 4048  E40A  3C F6                               .WORD   NOFUNC-1     ;(SPECIAL)
 4049  E40C  4C E4 F3                            JMP     PWRONA
 4050  E40F  00                                 .BYTE   0            ;ROM FILLER BYTE
 4051                                    ;
 4052                                        *=SCRENV
 4053                                    ;
 4054                                    ; DISPLAY HANDLER ENTRY POINT
 4055                                    ;
 4056  E410  F5 F3                       DISPLA: .WORD   DOPEN-1
 4057  E412  33 F6                               .WORD   RETUR1-1     ;(CLOSE)
 4058  E414  92 F5                               .WORD   GETCH-1
 4059  E416  B6 F5                               .WORD   OUTCH-1
 4060  E418  33 F6                               .WORD   RETUR1-1     ;(STATUS)
 4061  E41A  FB FC                               .WORD   DRAW-1       ;(SPECIAL)
 4062  E41C  4C E4 F3                            JMP     PWRONA
 4063  E41F  00                                 .BYTE   0            ;ROM FILLER BYTE
 4064                                    ;
 4065                                        *=KEYBDV
 4066                                    ;
 4067                                    ;
 4068                                    ; KEYBOARD HANDLER ENTRY POINT
 4069                                    ;
 4070  E420  33 F6                       KBDHND: .WORD   RETUR1-1
 4071  E422  33 F6                               .WORD   RETUR1-1     ;(CLOSE)
 4072  E424  E1 F6                               .WORD   KGETCH-1
 4073  E426  3C F6                               .WORD   NOFUNC-1     ;(OUTCH)
 4074  E428  33 F6                               .WORD   RETUR1-1     ;(STATUS)
 4075  E42A  3C F6                               .WORD   NOFUNC-1     ;(SPECIAL)
 4076  E42C  4C E4 F3                            JMP     PWRONA
 4077  E42F  00                                 .BYTE   0            ;ROM FILLER BYTE
 4078                                    ;
 4079                                    ;
 4080                                    ; INTERRUPT VECTOR TABLE ENTRY
 4081                                        *=VCTABL-INTABS+VKEYBD
 4082  E488  BE FF                           .WORD   PIRQS        ;KEYBOARD IRQ INTERRUPT VECTOR
```

```
4083
4084                    *=KBDORG
4085                ;
4086  F3E4  A9 FF  PWRONA: LDA     #$FF
4087  F3E6  8D FC 02        STA     CH
4088  F3E9  AD E6 02        LDA     MEMTOP+1
4089  F3EC  29 F0           AND     #$F0        ;INSURE 4K PAGE BOUNDARY
4090  F3EE  85 6A           STA     RAMTOP
4091  F3F0  A9 40           LDA     #$40        ;DEFAULT TO UPPER CASE ALPHA AT PWRON
4092  F3F2  8D BE 02        STA     SHFLOK
4093  F3F5  60              RTS                 ;POWER ON COMPLETED
```

```
4094                            .PAGE
4095                    ;
4096                    ;
4097                    ; BEGIN DISPLAY HANDLER OPEN PROCESSING
4098                    ;
4099  F3F6  A5 2B       DOPEN:  LDA     ICAX2Z          ;GET AUX 2 BYTE
4100  F3F8  29 0F               AND     #$F
4101  F3FA  D0 08               BNE     OPNCOM          ;IF MODE ZERO, CLEAR ICAX1Z
4102  F3FC  A5 2A       EOPEN:  LDA     ICAX1Z          ;CLEAR "CLR INHIBIT" AND "MXD MODE" BITS
4103  F3FE  29 0F               AND     #$F
4104  F400  85 2A               STA     ICAX1Z
4105  F402  A9 00               LDA     #0
4106  F404  85 57       OPNCOM: STA     DINDEX
4107  F406  A9 E0               LDA     #$E0            ;INITIALIZE GLOBAL VBLANK RAM
4108  F408  8D F4 02            STA     CHBAS
4109  F40B  A9 02               LDA     #2
4110  F40D  8D F3 02            STA     CHACT
4111  F410  8D 2F 02            STA     SDMCTL          ;TURN OFF DMA NEXT VBLANK
4112  F413  A9 01               LDA     #SUCCES
4113  F415  85 4C               STA     DSTAT           ;CLEAR STATUS
4114  F417  A9 C0               LDA     #$C0            ;DO IRGEN
4115  F419  05 10               ORA     POKMSK
4116  F41B  85 10               STA     POKMSK
4117  F41D  8D 0E D2            STA     IRQEN
4118  F420  A9 00               LDA     #0
4119  F422  8D 93 02            STA     TINDEX          ;TEXT INDEX MUST ALWAYS BE 0
4120  F425  85 64               STA     ADRESS
4121  F427  85 7B               STA     SWPFLG
4122  F429  8D F0 02            STA     CRSINH          ;TURN CURSOR ON AT OPEN
4123  F42C  A0 0E               LDY     #14             ;CLEAR TAB STOPS
4124  F42E  A9 01               LDA     #1              ;INIT TAB STOPS TO EVERY 8 CHARACTERS
4125  F430  99 A3 02    CLRTBS: STA     TABMAP,Y
4126  F433  88                  DEY
4127  F434  10 FA               BPL     CLRTBS
4128  F436  A2 04               LDX     #4              ;LOAD COLOR REGISTERS
4129  F438  BD C1 FE    DOPEN8: LDA     COLRTB,X
4130  F43B  9D C4 02            STA     COLOR0,X
4131  F43E  CA                  DEX
4132  F43F  10 F7               BPL     DOPEN8
4133  F441  A4 6A               LDY     RAMTOP          ;DO TXTMSC=$2C40 (IF MEMTOP=3000)
4134  F443  88                  DEY
4135  F444  8C 95 02            STY     TXTMSC+1
4136  F447  A9 60               LDA     #$60
4137  F449  8D 94 02            STA     TXTMSC
4138  F44C  A6 57               LDX     DINDEX
4139  F44E  BD 69 FE            LDA     ANCONV,X        ;CONVERT IT TO ANTIC CODE
4140  F451  D0 04               BNE     DOPENA          ;IF ZERO, IT IS ILLEGAL
4141  F453  A9 91       OPNERR: LDA     #BADMOD         ;SET ERROR STATUS
4142  F455  85 4C               STA     DSTAT
4143  F457  85 51       DOPENA: STA     HOLD1
4144  F459  A5 6A               LDA     RAMTOP          ;SET UP AN INDIRECT POINTER
4145  F45B  85 65               STA     ADRESS+1
4146  F45D  BC 45 FE            LDY     ALOCAT,X        ;ALLOCATE N BLOCKS OF 40 BYTES
4147  F460  A9 28       DOPEN1: LDA     #40
```

```
4148 F462 20 21 F9          JSR     DBSUB
4149 F465 88                DEY
4150 F466 D0 F8             BNE     DOPEN1
4151 F468 AD 6F 02          LDA     GPRIOR      ;CLEAR GTIA MODES
4152 F46B 29 3F             AND     #$3F
4153 F46D 85 67             STA     OPNTMP+1
4154 F46F A8                TAY
4155 F470 E0 08             CPX     #8          ;TEST IF 320X1
4156 F472 90 17             BCC     NOT8
4157 F474 8A                TXA                 ;GET 2 LOW BITS
4158 F475 6A                ROR     A
4159 F476 6A                ROR     A
4160 F477 6A                ROR     A
4161 F478 29 C0             AND     #$C0        ;NOW 2 TOP BITS
4162 F47A 05 67             ORA     OPNTMP+1
4163 F47C A8                TAY
4164 F47D A9 10             LDA     #16         ;SUBTRACT 16 MORE FOR PAGE BOUNDARY
4165 F47F 20 21 F9          JSR     DBSUB
4166 F482 E0 0B             CPX     #11         ;TEST MODE 11
4167 F484 D0 05             BNE     NOT8        ;IF MODE = 11
4168 F486 A9 06             LDA     #6          ;PUT GTIA LUM VALUE INTO BACKGROUND REGISTER
4169 F488 8D C8 02          STA     COLOR4
4170 F48B 8C 6F 02  NOT8:   STY     GPRIOR      ;STORE NEW PRIORITY
4171 F48E A5 64             LDA     ADRESS      ;SAVE MEMORY SCAN COUNTER ADDRESS
4172 F490 85 58             STA     SAVMSC
4173 F492 A5 65             LDA     ADRESS+1
4174 F494 85 59             STA     SAVMSC+1
4175 F496 AD 0B D4  VBWAIT: LDA     VCOUNT      ;WAIT FOR NEXT VBLANK BEFORE MESSING
4176 F499 C9 7A             CMP     #$7A        ;WITH THE DISPLAY LIST
4177 F49B D0 F9             BNE     VBWAIT
4178 F49D 20 1F F9          JSR     DBDEC       ;START PUTTING DISPLAY LIST RIGHT UNDER RAM
4179 F4A0 BD 75 FE          LDA     PAGETB,X    ;TEST IF DISPLAY LIST WILL BE IN TROUBLE
4180 F4A3 F0 06             BEQ     NOMOD       ;OF CROSSING A 256 BYTE PAGE BOUNDARY
4181 F4A5 A9 FF             LDA     #$FF        ;IF SO, DROP DOWN A PAGE
4182 F4A7 85 64             STA     ADRESS
4183 F4A9 C6 65             DEC     ADRESS+1
4184 F4AB A5 64  NOMOD:     LDA     ADRESS      ;SAVE END OF DISPLAY LIST FOR LATER
4185 F4AD 85 68             STA     SAVADR
4186 F4AF A5 65             LDA     ADRESS+1
4187 F4B1 85 69             STA     SAVADR+1
4188 F4B3 20 13 F9          JSR     DBDDEC      ;(DOUBLE BYTE DOUBLE DECREMENT)
4189 F4B6 A9 41             LDA     #$41        ;(ANTIC) WAIT FOR VBLANK AND JMP TO TOP
4190 F4B8 20 17 F9          JSR     STORE
4191 F4BB 86 66             STX     OPNTMP
4192 F4BD A9 18             LDA     #24         ;INITIALIZE BOTSCR
4193 F4BF 8D BF 02          STA     BOTSCR
4194 F4C2 A5 57             LDA     DINDEX      ;DISALLOW MIXED MODE IF MODE.GE.9
4195 F4C4 C9 09             CMP     #9
4196 F4C6 B0 2D             BCS     NOTMXD
4197 F4C8 A5 2A             LDA     ICAX1Z      ;TEST MIXED MODE
4198 F4CA 29 10             AND     #$10
4199 F4CC F0 27             BEQ     NOTMXD
4200 F4CE A9 04             LDA     #4
4201 F4D0 8D BF 02          STA     BOTSCR
```

```
4202 F4D3 A2 02          LDX   #2          ;ADD 4 LINES OF TEXT AT BOTTOM OF SCREEN
4203 F4D5 A9 02 DOPEN2:  LDA   #2
4204 F4D7 20 17 F9       JSR   STORE
4205 F4DA CA             DEX
4206 F4DB 10 F8          BPL   DOPEN2
4207 F4DD A4 6A          LDY   RAMTOP      ;RELOAD MSC FOR TEXT
4208 F4DF 88             DEY
4209 F4E0 98             TYA
4210 F4E1 20 17 F9       JSR   STORE
4211 F4E4 A9 60          LDA   #$60
4212 F4E6 20 17 F9       JSR   STORE
4213 F4E9 A9 42          LDA   #$42
4214 F4EB 20 17 F9       JSR   STORE
4215 F4EE 18             CLC
4216 F4EF A9 0C          LDA   #MXDMDE-NUMDLE ;POINT X AT MIXED MODE TABLE
4217 F4F1 65 66          ADC   OPNTMP
4218 F4F3 85 66          STA   OPNTMP
4219 F4F5 A4 66 NOTMXD:  LDY   OPNTMP
4220 F4F7 BE 51 FE       LDX   NUMDLE,Y    ;GET NUMBER OF DISPLAY LIST ENTRIES
4221 F4FA A5 51 DOPEN3:  LDA   HOLD1       ;STORE N DLE'S
4222 F4FC 20 17 F9       JSR   STORE
4223 F4FF CA             DEX
4224 F500 D0 F8          BNE   DOPEN3
4225 F502 A5 57          LDA   DINDEX      ;DO THE MESSY 320X1 PROBLEM
4226 F504 C9 08          CMP   #8
4227 F506 90 1C          BCC   DOPEN5
4228 F508 A2 5D          LDX   #93         ;GET REMAINING NUMBER OF DLE'S
4229 F50A A5 6A          LDA   RAMTOP      ;RELOAD MEMORY SCAN COUNTER
4230 F50C 38             SEC
4231 F50D E9 10          SBC   #$10
4232 F50F 20 17 F9       JSR   STORE
4233 F512 A9 00          LDA   #0
4234 F514 20 17 F9       JSR   STORE
4235 F517 A9 4F          LDA   #$4F        ;(ANTIC) RELOAD MSC CODE
4236 F519 20 17 F9       JSR   STORE
4237 F51C A5 51 DOPEN4:  LDA   HOLD1       ;DO REMAINING DLE'S
4238 F51E 20 17 F9       JSR   STORE
4239 F521 CA             DEX
4240 F522 D0 F8          BNE   DOPEN4
4241 F524 A5 59 DOPEN5:  LDA   SAVMSC+1    ;POLISH OFF DISPLAY LIST
4242 F526 20 17 F9       JSR   STORE
4243 F529 A5 58          LDA   SAVMSC
4244 F52B 20 17 F9       JSR   STORE
4245 F52E A5 51          LDA   HOLD1
4246 F530 09 40          ORA   #$40
4247 F532 20 17 F9       JSR   STORE
4248 F535 A9 70          LDA   #$70        ;24 BLANK LINES
4249 F537 20 17 F9       JSR   STORE
4250 F53A A9 70          LDA   #$70
4251 F53C 20 17 F9       JSR   STORE
4252 F53F A5 64          LDA   ADRESS      ;SAVE DISPLAY LIST ADDRESS
4253 F541 8D 30 02       STA   SDLSTL
4254 F544 A5 65          LDA   ADRESS+1
4255 F546 8D 31 02       STA   SDLSTL+1
```

```
4256  F549  A9 70          LDA   #$70        ;ADD LAST BLANK LINE ENTRY
4257  F54B  20 17 F9       JSR   STORE       ;POSITION ADRESS=SDLSTL-1
4258  F54E  A5 64          LDA   ADRESS      ;STORE NEW MEMTOP
4259  F550  8D E5 02       STA   MEMTOP
4260  F553  A5 65          LDA   ADRESS+1
4261  F555  8D E6 02       STA   MEMTOP+1
4262  F558  A5 68          LDA   SAVADR
4263  F55A  85 64          STA   ADRESS
4264  F55C  A5 69          LDA   SAVADR+1
4265  F55E  85 65          STA   ADRESS+1
4266  F560  AD 31 02       LDA   SDLSTL+1
4267  F563  20 17 F9       JSR   STORE
4268  F566  AD 30 02       LDA   SDLSTL
4269  F569  20 17 F9       JSR   STORE
4270  F56C  A5 4C          LDA   DSTAT       ;IF ERROR OCURRED ON ALLOCATION, OPEN THE ED
4271  F56E  10 07          BPL   DOPEN9
4272  F570  48             PHA               ;SAVE STATUS
4273  F571  20 FC F3       JSR   EOPEN       ;OPEN THE EDITOR
4274  F574  68             PLA               ;RESTORE STATUS
4275  F575  A8             TAY               ;AND RETURN IT TO CIO
4276  F576  60             RTS
4277  F577  A5 2A    DOPEN9: LDA   ICAX1Z     ;TEST CLEAR INHIBIT BIT
4278  F579  29 20          AND   #$20
4279  F57B  D0 0B          BNE   DOPEN7
4280  F57D  20 B9 F7       JSR   CLRSCR      ;CLEAR SCREEN
4281  F580  8D 90 02       STA   TXTROW      ;AND HOME TEXT CURSOR (AC IS ZERO)
4282  F583  A5 52          LDA   LMARGN
4283  F585  8D 91 02       STA   TXTCOL
4284  F588  A9 22    DOPEN7: LDA   #$22        ;EVERYTHING ELSE IS SET UP
4285  F58A  0D 2F 02       ORA   SDMCTL      ;SO TURN ON DMACTL
4286  F58D  8D 2F 02       STA   SDMCTL
4287  F590  4C 21 F6       JMP   RETUR2
4288                ;
4289                ;
4290  F593  20 96 FA  GETCH: JSR   RANGE       ;GETCH DOES INCRSR, GETPLT DOESN'T
4291  F596  20 A2 F5       JSR   GETPLT
4292  F599  20 32 F8       JSR   INATAC      ;CONVERT INTERNAL CODE TO ATASCII
4293  F59C  20 D4 F9       JSR   INCRSB
4294  F59F  4C 34 F6       JMP   RETUR1
4295  F5A2  20 47 F9  GETPLT: JSR   CONVRT     ;CONVERT ROW/COLUMN TO ADRESS
4296  F5A5  B1 64          LDA   (ADRESS),Y
4297  F5A7  2D A0 02       AND   DMASK
4298  F5AA  46 6F    SHIFTD: LSR   SHFAMT      ;SHIFT DATA DOWN TO LOW BITS
4299  F5AC  B0 03          BCS   SHIFT1
4300  F5AE  4A             LSR   A
4301  F5AF  10 F9          BPL   SHIFTD      ;(UNCONDITIONAL)
4302  F5B1  8D FA 02  SHIFT1: STA   CHAR
4303  F5B4  C9 00          CMP   #0          ;RESTORE FLAGS ALSO
4304  F5B6  60             RTS
4305                ;
4306                ;
4307                ;
4308  F5B7  8D FB 02  OUTCH: STA   ATACHR
4309  F5BA  20 96 FA       JSR   RANGE
```

```
4310                        ;       JSR     OFFCRS
4311  F5BD  AD FB 02        OUTCHA: LDA     ATACHR      ;TEST FOR CLEAR SCREEN
4312  F5C0  C9 7D                   CMP     #CLRCOD
4313  F5C2  D0 06                   BNE     OUTCHE
4314  F5C4  20 B9 F7                JSR     CLRSCR
4315  F5C7  4C 21 F6                JMP     RETUR2
4316  F5CA  AD FB 02        OUTCHE: LDA     ATACHR      ;TEST FOR CARRIAGE RETURN
4317  F5CD  C9 9B                   CMP     #CR
4318  F5CF  D0 06                   BNE     OUTCHB
4319  F5D1  20 30 FA                JSR     DOCRWS      ;DO CR
4320  F5D4  4C 21 F6                JMP     RETUR2
4321  F5D7  20 E0 F5        OUTCHB: JSR     OUTPLT
4322  F5DA  20 D8 F9                JSR     INCRSR
4323  F5DD  4C 21 F6                JMP     RETUR2
4324                        ;
4325                        ;
4326  F5E0  AD FF 02        OUTPLT: LDA     SSFLAG      ;*****LOOP HERE IF START/STOP FLAG IS NON-0
4327  F5E3  D0 FB                   BNE     OUTPLT
4328  F5E5  A2 02                   LDX     #2
4329  F5E7  B5 54           CRLOOP: LDA     ROWCRS,X    ;SAVE CURSOR LOCATION FOR DRAW LINE TO DRAW
4330  F5E9  95 5A                   STA     OLDROW,X
4331  F5EB  CA                      DEX
4332  F5EC  10 F9                   BPL     CRLOOP
4333  F5EE  AD FB 02                LDA     ATACHR      ;CONVERT ATASCII(ATACHR) TO INTERNAL(CHAR)
4334  F5F1  A8                      TAY                 ;SAVE ATACHR
4335  F5F2  2A                      ROL     A
4336  F5F3  2A                      ROL     A
4337  F5F4  2A                      ROL     A
4338  F5F5  2A                      ROL     A
4339  F5F6  29 03                   AND     #3
4340  F5F8  AA                      TAX                 ;X HAS INDEX INTO ATAINT
4341  F5F9  98                      TYA                 ;RESTORE ATACHR
4342  F5FA  29 9F                   AND     #$9F        ;STRIP OFF COLUMN ADDRESS
4343  F5FC  1D F6 FE                ORA     ATAINT,X    ;OR IN NEW COLUMN ADDRESS
4344  F5FF  8D FA 02        OUTCH2: STA     CHAR
4345  F602  20 47 F9                JSR     CONVRT
4346  F605  AD FA 02                LDA     CHAR
4347  F608  46 6F           SHIFTU: LSR     SHFAMT      ;SHIFT UP TO PROPER POSITION
4348  F60A  B0 04                   BCS     SHIFT2
4349  F60C  0A                      ASL     A
4350  F60D  4C 08 F6                JMP     SHIFTU
4351  F610  2D A0 02        SHIFT2: AND     DMASK
4352  F613  85 50                   STA     TMPCHR      ;SAVE SHIFTED DATA
4353  F615  AD A0 02                LDA     DMASK       ;INVERT MASK
4354  F618  49 FF                   EOR     #$FF
4355  F61A  31 64                   AND     (ADRESS),Y  ;MASK OFF OLD DATA
4356  F61C  05 50                   ORA     TMPCHR      ;OR IN NEW DATA
4357  F61E  91 64                   STA     (ADRESS),Y
4358  F620  60                      RTS
4359                        ;
4360                        ;
4361  F621  20 A2 F5        RETUR2: JSR     GETPLT      ;DO CURSOR ON THE WAY OUT
4362  F624  85 5D                   STA     OLDCHR
4363  F626  A6 57                   LDX     DINDEX      ;GRAPHICS HAVE INVISIBLE CURSOR
```

```
4364  F628 D0 0A          BNE     RETUR1
4365  F62A AE F0 02       LDX     CRSINH      ;TEST CURSOR INHIBIT
4366  F62D D0 05          BNE     RETUR1
4367  F62F 49 80          EOR     #$80        ;TOGGLE MSB
4368  F631 20 FF F5       JSR     OUTCH2      ;DISPLAY IT
4369  F634 A4 4C    RETUR1: LDY   DSTAT       ;RETURN TO CIO WITH STATUS IN Y
4370  F636 A9 01          LDA     #SUCCES
4371  F638 85 4C          STA     DSTAT       ;SET STATUS= SUCCESSFUL COMPLETION
4372  F63A AD FB 02       LDA     ATACHR      ;PUT ATACHR IN AC FOR RETURN TO CIO
4373  F63D 60       NOFUNC: RTS               ;(NON-EXISTENT FUNCTION RETURN POINT)
4374                ;
4375                ;
4376                ;
4377                ; END OF DISPLAY HANDLER
4378                ;
```

```
4379                                    PAGE
4380
4381
4382
4383
4384  F63E  20 B3 FC    EGETCH: JSR     SWAP
4385  F641  20 88 FA            JSR     ERANGE
4386  F644  A5 6B               LDA     BUFCNT      ;ANYTHING IN THE BUFFER?
4387  F646  D0 34               BNE     EGETC3      ;YES
4388  F648  A5 54               LDA     ROWCRS      ;NO, SO SAVE BUFFER START ADDRESS
4389  F64A  85 6C               STA     BUFSTR
4390  F64C  A5 55               LDA     COLCRS
4391  F64E  85 6D               STA     BUFSTR+1
4392  F650  20 E2 F6    EGETC1: JSR     KGETCH ;LET'S FILL OUR BUFFER
4393  F653  84 4C               STY     DSTAT       SAVE KEYBOARD STATUS
4394  F655  AD FB 02            LDA     ATACHR      ;TEST FOR CR
4395  F658  C9 9B               CMP     #CR
4396  F65A  F0 12               BEQ     EGETC2
4397  F65C  20 AD F6            JSR     DOSS        ;NO, SO PRINT IT
4398  F65F  20 B3 FC            JSR     SWAP        ;JSR DOSS DID SWAP SO SWAP BACK
4399  F662  A5 63               LDA     LOGCOL      ;BEEP IF NEARING LOGICAL COL 120
4400  F664  C9 71               CMP     #113
4401  F666  D0 03               BNE     EGETC6
4402  F668  20 0A F9            JSR     BELL
4403  F66B  4C 50 F6    EGETC6: JMP     EGETC1
4404  F66E  20 E4 FA    EGETC2: JSR     OFFCRS      ;GET BUFFER COUNT.
4405  F671  20 00 FC            JSR     DOBUFC
4406  F674  A5 6C               LDA     BUFSTR      ;RETURN A CHARACTER
4407  F676  85 54               STA     ROWCRS
4408  F678  A5 6D               LDA     BUFSTR+1
4409  F67A  85 55               STA     COLCRS
4410  F67C  A5 6B       EGETC3: LDA     BUFCNT
4411  F67E  F0 11               BEQ     EGETC5
4412  F680  C6 6B       EGETC7: DEC     BUFCNT      ;AND RETURN TILL BUFCNT=0
4413  F682  F0 0D               BEQ     EGETC5
4414  F684  A5 4C               LDA     DSTAT       ;IF ERR, LOOP ON EGETC7 UNTIL BUFR IS EMPTIE
4415  F686  30 F8               BMI     EGETC7
4416  F688  20 93 F5            JSR     GETCH
4417  F68B  8D FB 02            STA     ATACHR
4418  F68E  4C B3 FC            JMP     SWAP        ;AND RETURN WITHOUT TURNING CURSOR BACK ON
4419  F691  20 30 FA    EGETC5: JSR     DOCRWS      ;DO REAL CARRIAGE RETURN
4420  F694  A9 9B               LDA     #CR         ;AND RETURN EOL
4421  F696  8D FB 02            STA     ATACHR
4422  F699  20 21 F6            JSR     RETUR2      ;TURN ON CURSOR THEN SWAP
4423  F69C  84 4C               STY     DSTAT       ;SAVE KEYBOARD STATUS
4424  F69E  4C B3 FC            JMP     SWAP        ;AND RETURN THROUGH RETUR1
4425                    ;
4426  F6A1  6C 64 00    JSRIND: JMP     (ADRESS)    ;JSR TO THIS CAUSES JSR INDIRECT
4427                    ;
4428  F6A4  8D FB 02    EOUTCH: STA     ATACHR      ;SAVE ATASCII VALUE
4429  F6A7  20 B3 FC            JSR     SWAP
4430  F6AA  20 88 FA            JSR     ERANGE
4431  F6AD  20 E4 FA    DOSS:   JSR     OFFCRS      ;TURN OFF CURSOR
4432  F6B0  20 8D FC            JSR     TSTCTL      ;TEST FOR CONTROL CHARACTERS (Z=1 IF CTL)
```

```
4433 F6B3 F0 09          BEQ     EOUTC5
4434 F6B5 0E A2 02 EOUTC6: ASL   ESCFLG      ;ESCFLG ONLY WORKS ONCE
4435 F6B8 20 CA F5       JSR     OUTCHE
4436 F6BB 4C B3 FC ERETN: JMP    SWAP        ;AND RETURN THROUGH RETUR1
4437 F6BE AD FE 02 EOUTC5: LDA   DSPFLG      ;DO DSPFLG AND ESCFLG
4438 F6C1 0D A2 02       ORA     ESCFLG
4439 F6C4 D0 EF          BNE     EOUTC6      ;IF NON-0 DISPLAY RATHER THAN EXECUTE IT
4440 F6C6 0E A2 02       ASL     ESCFLG
4441 F6C9 E8             INX                 ;PROCESS CONTROL CHARACTERS
4442 F6CA BD C6 FE       LDA     CNTRLS,X    ;GET DISPLACEMENT INTO ROUTINE
4443 F6CD 85 64          STA     ADRESS
4444 F6CF BD C7 FE       LDA     CNTRLS+1,X  ;GET HIGH BYTE
4445 F6D2 85 65          STA     ADRESS+1
4446 F6D4 20 A1 F6       JSR     JSRIND      ;DO COMPUTED JSR
4447 F6D7 20 21 F6       JSR     RETUR2      ;DO CURSOR
4448 F6DA 4C B3 FC       JMP     SWAP        ;ALL DONE SO RETURN THROUGH RETUR1
4449                  ;
4450                  ;
4451                  ;
4452                  ;
4453                  ; END SCREEN EDITOR.
4454                  ;
4455                  ;
4456                  ; BEGIN KEYBOARD HANDLER
4457                  ;
4458                  ;
4459                  ;
4460                  ;
4461 F6DD A9 FF       KGETC2: LDA     #$FF
4462 F6DF 8D FC 02       STA     CH
4463 F6E2 A5 2A       KGETCH: LDA     ICAX1Z      ;TEST LSB OF AUX1 FOR SPECIAL EDITOR READ MO
4464 F6E4 4A             LSR     A
4465 F6E5 B0 62          BCS     GETOUT
4466 F6E7 A9 80          LDA     #BRKABT
4467 F6E9 A6 11          LDX     BRKKEY      ;TEST BREAK
4468 F6EB F0 58          BEQ     K7          ;IF BREAK, PUT BRKABT IN DSTAT AND CR IN ATA
4469 F6ED AD FC 02       LDA     CH
4470 F6F0 C9 FF          CMP     #$FF
4471 F6F2 F0 EE          BEQ     KGETCH
4472 F6F4 85 7C          STA     HOLDCH      ;SAVE CH FOR SHIFT LOCK PROC
4473 F6F6 A2 FF          LDX     #$FF        ;"CLEAR" CH
4474 F6F8 8E FC 02       STX     CH
4475 F6FB 20 D8 FC       JSR     CLICK       ;DO KEYBOARD AUDIO FEEDBACK (A IS OK)
4476 F6FE AA          KGETC3: TAX             ;DO ASCCON
4477 F6FF E0 C0          CPX     #$C0        ;TEST FOR CTL & SHIFT TOGETHER
4478 F701 90 02          BCC     ASCC01
4479 F703 A2 03          LDX     #3          ;BAD CODE
4480 F705 BD FE FE ASCC01: LDA   ATASCI,X
4481 F708 8D FB 02       STA     ATACHR      ;DONE
4482 F70B C9 80          CMP     #$80        ;DO NULLS
4483 F70D F0 CE          BEQ     KGETC2
4484 F70F C9 81          CMP     #$81        ;CHECK ATARI KEY
4485 F711 D0 0B          BNE     KGETC1
4486 F713 AD B6 02       LDA     INVFLG
```

```
4487 F716 49 80          EOR   #$80
4488 F718 8D B6 02       STA   INVFLG
4489 F71B 4C DD F6       JMP   KGETC2      ;DONT RETURN A VALUE
4490 F71E C9 82    KGETC1: CMP   #$82       ;CAPS/LOWER
4491 F720 D0 07          BNE   K1
4492 F722 A9 00          LDA   #0          ;CLEAR SHFLOK
4493 F724 8D BE 02       STA   SHFLOK
4494 F727 F0 B4          BEQ   KGETC2
4495 F729 C9 83    K1:   CMP   #$83        ;SHIFT CAPS/LOWER
4496 F72B D0 07          BNE   K2
4497 F72D A9 40          LDA   #$40
4498 F72F 8D BE 02       STA   SHFLOK      ;SHIFT BIT
4499 F732 D0 A9          BNE   KGETC2
4500 F734 C9 84    K2:   CMP   #$84        ;CNTL CAPS/LOWER
4501 F736 D0 07          BNE   K3
4502 F738 A9 80          LDA   #$80        ;CNTL BIT
4503 F73A 8D BE 02       STA   SHFLOK
4504 F73D D0 9E          BNE   KGETC2
4505 F73F C9 85    K3:   CMP   #$85        ;DO EOF
4506 F741 D0 0A          BNE   K6
4507 F743 A9 88          LDA   #EOFERR
4508 F745 85 4C    K7:   STA   DSTAT
4509 F747 85 11          STA   BRKKEY      ;RESTORE BREAK
4510 F749 A9 9B    GETOUT: LDA   #CR       ;PUT CR IN ATACHR
4511 F74B D0 26          BNE   K8          ;(UNCONDITIONAL)
4512 F74D A5 7C    K6:   LDA   HOLDCH      ;PROCESS SHIFT LOCKS
4513 F74F C9 40          CMP   #$40        ;REGULAR SHIFT AND CONTROL TAKE PRECEDENCE
4514 F751 B0 15          BCS   K5          ;OVER LOCK
4515 F753 AD FB 02       LDA   ATACHR      ;TEST FOR ALPHA
4516 F756 C9 61          CMP   #$61        ;LOWER CASE A
4517 F758 90 0E          BCC   K5          ;NOT ALPHA IF LT
4518 F75A C9 7B          CMP   #$7B        ;LOWER CASE Z+1
4519 F75C B0 0A          BCS   K5          ;NOT ALPHA IF GE
4520 F75E AD BE 02       LDA   SHFLOK      ;DO SHIFT/CONTROL LOCK
4521 F761 F0 05          BEQ   K5          ;IF NO LOCK, DONT RE-DO IT
4522 F763 05 7C          ORA   HOLDCH
4523 F765 4C FE F6       JMP   KGETC3      ;DO RETRY
4524 F768 20 8D FC K5:   JSR   TSTCTL      ;DONT INVERT MSB OF CONTROL CHARACTERS
4525 F76B F0 09          BEQ   K4
4526 F76D AD FB 02       LDA   ATACHR
4527 F770 4D B6 02       EOR   INVFLG
4528 F773 8D FB 02 K8:   STA   ATACHR
4529 F776 4C 34 F6 K4:   JMP   RETUR1      ALL DONE
4530                     ;
4531                     ;
```

```
4532                            .PAGE
4533                    ;
4534                    ;
4535                    ; CONTROL CHARACTER PROCESSORS
4536                    ;
4537 F779 A9 80        ESCAPE: LDA     #$80            ;SET ESCAPE FLAG
4538 F77B 8D A2 02              STA     ESCFLG
4539 F77E 60                    RTS
4540 F77F C6 54        CRSRUP: DEC     ROWCRS
4541 F781 10 06                 BPL     COMRET
4542 F783 AE BF 02              LDX     BOTSCR          ;WRAPAROUND
4543 F786 CA                    DEX
4544 F787 86 54        UPDNCM: STX     ROWCRS
4545 F789 4C 5C FC     COMRET: JMP     STRBEG          ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4546 F78C E6 54        CRSRDN: INC     ROWCRS
4547 F78E A5 54                LDA     ROWCRS
4548 F790 CD BF 02             CMP     BOTSCR
4549 F793 90 F4                BCC     COMRET
4550 F795 A2 00                LDX     #0
4551 F797 F0 EE                BEQ     UPDNCM          ;(UNCONDITIONAL)
4552 F799 C6 55        CRSRLF: DEC     COLCRS
4553 F79B A5 55                LDA     COLCRS
4554 F79D 30 04                BMI     CRSRL1          ;(IF LMARGN=0, THIS ELIMINATES PROBLEM CASE)
4555 F79F C5 52                CMP     LMARGN
4556 F7A1 B0 04                BCS     COMRE1
4557 F7A3 A5 53        CRSRL1: LDA     RMARGN
4558 F7A5 85 55        LFRTCM: STA     COLCRS
4559 F7A7 4C DD FB     COMRE1: JMP     DOLCOL          ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4560 F7AA E6 55        CRSRRT: INC     COLCRS
4561 F7AC A5 55                LDA     COLCRS
4562 F7AE C5 53                CMP     RMARGN
4563 F7B0 90 F5                BCC     COMRE1
4564 F7B2 F0 F3                BEQ     COMRE1          ;(CAUSE BLE)
4565 F7B4 A5 52                LDA     LMARGN
4566 F7B6 4C A5 F7             JMP     LFRTCM          ;UNCONDITIONAL TO COMMON STORE
4567 F7B9 20 F3 FC     CLRSCR: JSR     PUTMSC
4568 F7BC A0 00                LDY     #0
4569 F7BE 98                   TYA                     ;PUT 0 IN THE AC
4570 F7BF 91 64        CLRSC2: STA     (ADRESS),Y      ;(AC IS ZERO)
4571 F7C1 C8                   INY
4572 F7C2 D0 FB                BNE     CLRSC2
4573 F7C4 E6 65                INC     ADRESS+1
4574 F7C6 A6 65                LDX     ADRESS+1
4575 F7C8 E4 6A                CPX     RAMTOP
4576 F7CA 90 F3                BCC     CLRSC2
4577 F7CC A9 FF                LDA     #$FF            ;CLEAN UP LOGICAL LINE BIT MAP
4578 F7CE 99 B2 02     CLRSC3: STA     LOGMAP,Y        ;(Y IS ZERO AFTER CLRSC2 LOOP)
4579 F7D1 C8                   INY
4580 F7D2 C0 04                CPY     #4
4581 F7D4 90 F8                BCC     CLRSC3
4582 F7D6 20 E4 FC     HOME:   JSR     COLCR           ;PLACE COLCRS AT LEFT EDGE
4583 F7D9 85 63                STA     LOGCOL
4584 F7DB 85 6D                STA     BUFSTR+1
4585 F7DD A9 00                LDA     #0
```

```
4586  F7DF  85 54          STA     ROWCRS
4587  F7E1  85 56          STA     COLCRS+1
4588  F7E3  85 6C          STA     BUFSTR
4589  F7E5  60             RTS
4590                       ;
4591  F7E6  A5 63          BS:     LDA     LOGCOL      ;BACKSPACE
4592  F7E8  C5 52                  CMP     LMARGN
4593  F7EA  F0 21                  BEQ     BS1
4594  F7EC  A5 55          BSA:    LDA     COLCRS      ;LEFT EDGE?
4595  F7EE  C5 52                  CMP     LMARGN
4596  F7F0  D0 03                  BNE     BS3         ;NO
4597  F7F2  20 73 FC               JSR     DELTIM      ;YES, SEE IF LINE SHOULD BE DELETED
4598  F7F5  20 99 F7       BS3:    JSR     CRSRLF
4599  F7F8  A5 55                  LDA     COLCRS
4600  F7FA  C5 53                  CMP     RMARGN
4601  F7FC  D0 07                  BNE     BS2
4602  F7FE  A5 54                  LDA     ROWCRS
4603  F800  F0 03                  BEQ     BS2
4604  F802  20 7F F7               JSR     CRSRUP
4605  F805  A9 20          BS2:    LDA     #$20        ;MAKE BACKSPACE DESTRUCTIVE
4606  F807  8D FB 02               STA     ATACHR
4607  F80A  20 E0 F5               JSR     OUTPLT
4608  F80D  4C DD FB       BS1:    JMP     DOLCOL      ;AND RETURN
4609  F810  20 AA F7       TAB:    JSR     CRSRRT      ;BEGIN SEARCH
4610  F813  A5 55                  LDA     COLCRS      ;TEST FOR NEW LINE
4611  F815  C5 52                  CMP     LMARGN
4612  F817  D0 0A                  BNE     TAB1        ;NO
4613  F819  20 34 FA               JSR     DOCR        ;DO CARRIAGE RETURN
4614  F81C  20 20 FB               JSR     LOGGET      ;CHECK IF END OF LOGICAL LINE
4615  F81F  90 02                  BCC     TAB1        ;NO, CONTINUE
4616  F821  B0 07                  BCS     TAB2        ;(UNCONDITIONAL)
4617  F823  A5 63          TAB1:   LDA     LOGCOL      ;CHECK FOR TAB STOP
4618  F825  20 25 FB               JSR     BITGET
4619  F828  90 E6                  BCC     TAB         ;NO, SO KEEP LOOKING
4620  F82A  4C DD FB       TAB2:   JMP     DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4621  F82D  A5 63          SETTAB: LDA     LOGCOL
4622  F82F  4C 06 FB               JMP     BITSET      ;SET BIT IN MAP AND RETURN
4623  F832  A5 63          CLRTAB: LDA     LOGCOL
4624  F834  4C 12 FB               JMP     BITCLR      ;CLEAR " " " " "
4625  F837  20 9D FC       INSCHR: JSR     PHACRS
4626  F83A  20 A2 F5               JSR     GETPLT      ;GET CHARACTER UNDER CURSOR
4627  F83D  85 7D                  STA     INSDAT
4628  F83F  A9 00                  LDA     #0
4629  F841  8D BB 02               STA     SCRFLG
4630  F844  20 FF F5       INSCH4: JSR     OUTCH2      ;STORE DATA
4631  F847  A5 63                  LDA     LOGCOL      ;SAVE LOGCOL: IF AFTER INCRSA LOGCOL IS
4632  F849  48                     PHA                 ;< THAN IT IS NOW, END LOOP
4633  F84A  20 DC F9               JSR     INCRSA      ;SPECIAL INCRSR ENTRY POINT
4634  F84D  68                     PLA
4635  F84E  C5 63                  CMP     LOGCOL
4636  F850  B0 0C                  BCS     INSCH3      ;QUIT
4637  F852  A5 7D          INSCH1: LDA     INSDAT      ;KEEP GOING
4638  F854  48                     PHA
4639  F855  20 A2 F5               JSR     GETPLT
```

```
4640  F858  85 7D          STA     INSDAT
4641  F85A  68             PLA
4642  F85B  4C 44 F8       JMP     INSCH4
4643  F85E  20 A8 FC INSCH3: JSR    PLACRS
4644  F861  CE BB 02 INSCH6: DEC    SCRFLG
4645  F864  30 04          BMI     INSCH5      ;IF SCROLL OCCURRED
4646  F866  C6 54          DEC     ROWCRS      ;MOVE CURSOR UP
4647  F868  D0 F7          BNE     INSCH6      ;(UNCOND) CONTINUE UNTIL SCRFLG IS MINUS
4648  F86A  4C DD FB INSCH5: JMP    DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4649                       ;
4650                       ;
4651  F86D  20 9D FC DELCHR: JSR    PHACRS
4652  F870  20 47 F9 DELCH1: JSR    CONVRT      ;GET DATA TO THE RIGHT OF THE CURSOR
4653  F873  A5 64          LDA     ADRESS
4654  F875  85 68          STA     SAVADR      ;SAVE ADRESS TO KNOW WHERE TO PUT DATA
4655  F877  A5 65          LDA     ADRESS+1
4656  F879  85 69          STA     SAVADR+1
4657  F87B  A5 63          LDA     LOGCOL
4658  F87D  48             PHA
4659  F87E  20 D4 F9       JSR     INCRSB      ;PUT CURSOR OVER NEXT CHARACTER
4660  F881  68             PLA
4661  F882  C5 63          CMP     LOGCOL      ;TEST NEW LOGCOL AGAINST OLD LOGCOL
4662  F884  B0 10          BCS     DELCH2      ;IF OLD.GE.NEW THEN QUIT
4663  F886  A5 54          LDA     ROWCRS      ;IS ROW OFF SCREEN?
4664  F888  CD BF 02       CMP     BOTSCR
4665  F88B  B0 09          BCS     DELCH2      ;YES, SO QUIT
4666  F88D  20 A2 F5       JSR     GETPLT      ;GET DATA UNDER CURSOR
4667  F890  A0 00          LDY     #0
4668  F892  91 68          STA     (SAVADR),Y  ;PUT IT IN PREVIOUS POSITION
4669  F894  F0 DA          BEQ     DELCH1      ;AND LOOP (UNCONDITIONAL)
4670  F896  A0 00 DELCH2: LDY     #0
4671  F898  98             TYA
4672  F899  91 68          STA     (SAVADR),Y  ;CLEAR THE LAST POSITION
4673  F89B  20 68 FC       JSR     DELTIA      ;TRY TO DELETE A LINE
4674  F89E  20 A8 FC       JSR     PLACRS
4675  F8A1  4C DD FB       JMP     DOLCOL      ;AND RETURN
4676  F8A4  38      INSLIN: SEC                ;NORMAL INSLIN PUTS "1" INTO BIT MAP
4677  F8A5  20 7B FB INSLIA: JSR    EXTEND      ;ENTRY POINT FOR C=0
4678  F8A8  A5 52          LDA     LMARGN      ;DO CARRIAGE RETURN (NO LF)
4679  F8AA  85 55          STA     COLCRS
4680  F8AC  20 47 F9       JSR     CONVRT      ;GET ADDRESS
4681  F8AF  A5 64          LDA     ADRESS      ;SET UP TO=40+FROM (FROM = CURSOR)
4682  F8B1  85 68          STA     FRMADR
4683  F8B3  18             CLC
4684  F8B4  69 28          ADC     #40
4685  F8B6  85 66          STA     TOADR
4686  F8B8  A5 65          LDA     ADRESS+1
4687  F8BA  85 69          STA     FRMADR+1
4688  F8BC  69 00          ADC     #0
4689  F8BE  85 67          STA     TOADR+1
4690  F8C0  A6 54          LDX     ROWCRS      ;SET UP LOOP COUNTER
4691  F8C2  E0 17          CPX     #23
4692  F8C4  F0 08          BEQ     INSLI2
4693  F8C6  20 4E FB INSLI1: JSR    MOVLIN
```

```
4694  F8C9  E8                INX
4695  F8CA  E0 17             CPX     #23
4696  F8CC  D0 F8             BNE     INSLI1
4697  F8CE  20 9B FB  INSLI2: JSR     CLRLIN      ;CLEAR CURRENT LINE
4698  F8D1  4C DD FB          JMP     DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4699  F8D4  20 DD FB  DELLIN: JSR     DOLCOL      ;GET BEGINNING OF LOG LINE (HOLD1)
4700  F8D7  A4 51     DELLIA: LDY     HOLD1       ;SQUEEZE BIT MAP
4701  F8D9  84 54             STY     ROWCRS      ;PUT CURSOR THERE
4702  F8DB  A4 54     DELLIB: LDY     ROWCRS
4703  F8DD  98        DELLI1: TYA
4704  F8DE  38                SEC
4705  F8DF  20 23 FB          JSR     LO2GET      ;GET NEXT BIT
4706  F8E2  08                PHP
4707  F8E3  98                TYA
4708  F8E4  18                CLC
4709  F8E5  69 78             ADC     #120
4710  F8E7  28                PLP
4711  F8E8  20 04 FB          JSR     BITPUT      ;WRITE IT OVER PRESENT BIT
4712  F8EB  C8                INY
4713  F8EC  C0 18             CPY     #24
4714  F8EE  D0 ED             BNE     DELLI1      ;LOOP
4715  F8F0  AD B4 02          LDA     LOGMAP+2    ;SET LSB
4716  F8F3  09 01             ORA     #1
4717  F8F5  8D B4 02          STA     LOGMAP+2
4718  F8F8  A5 52     DELLI2: LDA     LMARGN      ;DELETE LINE OF DATA USING PART OF SCROLL
4719  F8FA  85 55             STA     COLCRS      ;CR NO LF
4720  F8FC  20 47 F9          JSR     CONVRT
4721  F8FF  20 B7 FB          JSR     SCROL1
4722  F902  20 20 FB          JSR     LOGGET      ;TEST NEXT LINE FOR CONTINUATION
4723                  ; IS IT A NEW LOG LINE?
4724  F905  90 D4             BCC     DELLIB      ;NO SO DELETE ANOTHER
4725  F907  4C DD FB          JMP     DOLCOL      ;YES SO DOLCOL AND RETURN
4726  F90A  A0 20     BELL:   LDY     #$20
4727  F90C  20 D8 FC  BELL1:  JSR     CLICK
4728  F90F  88                DEY
4729  F910  10 FA             BPL     BELL1
4730  F912  60                RTS
```

```
4731                            .PAGE
4732                    ;
4733                    ;
4734                    ; ROUTINES
4735                    ;
4736                    ;
4737                    ; DOUBLE BYTE DECREMENT OF INDIRECT POINTER
4738                    ; INCLUDING DB SUBTRACT AND DB DOUBLE DECREMENT
4739                    ;
4740  F913  A9 02       DBDDEC: LDA     #2
4741  F915  D0 0A               BNE     DBSUB           ;(UNCONDITIONAL)
4742                    ;
4743                    ; STORE DATA INDIRECT AND DECREMENT POINTER
4744                    ; (PLACED HERE TO SAVE JMP DBDEC AFTER STORE)
4745  F917  A4 4C       STORE:  LDY     DSTAT           ;RETURN ON ERROR
4746  F919  30 2B               BMI     STROK
4747  F91B  A0 00               LDY     #0
4748  F91D  91 64       STORE1: STA     (ADRESS),Y
4749                    ;       JMP     DBDEC           DECREMENT AND RETURN
4750                    ;
4751  F91F  A9 01       DBDEC:  LDA     #1
4752  F921  8D 9E 02    DBSUB:  STA     SUBTMP
4753  F924  A5 4C               LDA     DSTAT           ;RETURN ON ERROR
4754  F926  30 1E               BMI     STROK
4755  F928  A5 64               LDA     ADRESS
4756  F92A  38                  SEC
4757  F92B  ED 9E 02            SBC     SUBTMP
4758  F92E  85 64               STA     ADRESS
4759  F930  B0 02               BCS     DBSUB1
4760  F932  C6 65               DEC     ADRESS+1
4761  F934  A5 0F       DBSUB1: LDA     APPMHI+1        ;MAKE SURE NOTHING EVER OVERWRITES APPMHI
4762  F936  C5 65               CMP     ADRESS+1
4763  F938  90 0C               BCC     STROK           ;OK
4764  F93A  D0 06               BNE     STRERR          ;ERROR
4765  F93C  A5 0E               LDA     APPMHI
4766  F93E  C5 64               CMP     ADRESS
4767  F940  90 04               BCC     STROK
4768  F942  A9 93       STRERR: LDA     #SCRMEM         ;SHOW MEM TOO SMALL FOR SCREEN ERROR
4769  F944  85 4C               STA     DSTAT
4770  F946  60          STROK:  RTS
4771                    ;
4772                    ;
4773                    ;
4774                    ; CONVERT ROW/COLUMN CURSOR INTO REAL ADDRESS (FROM SAVMSC ON UP)
4775                    ;
4776  F947  A5 54       CONVRT: LDA     ROWCRS          ;SAVE CURSOR
4777  F949  48                  PHA
4778  F94A  A5 55               LDA     COLCRS
4779  F94C  48                  PHA
4780  F94D  A5 56               LDA     COLCRS+1
4781  F94F  48                  PHA
4782  F950  20 F3 FC            JSR     PUTMSC
4783  F953  A5 54               LDA     ROWCRS          ;PUT 10*ROWCRS INTO MLTTMP
4784  F955  85 66               STA     MLTTMP
```

```
4785  F957  A9 00          LDA     #0
4786  F959  85 67          STA     MLTTMP+1
4787  F95B  A5 66          LDA     MLTTMP          ;QUICK X8
4788  F95D  0A             ASL     A
4789  F95E  26 67          ROL     MLTTMP+1
4790  F960  85 51          STA     HOLD1           ;(SAVE 2X VALUE)
4791  F962  A4 67          LDY     MLTTMP+1        ;""
4792  F964  8C 9F 02       STY     HOLD2           ;""
4793  F967  0A             ASL     A
4794  F968  26 67          ROL     MLTTMP+1
4795  F96A  0A             ASL     A
4796  F96B  26 67          ROL     MLTTMP+1
4797  F96D  18             CLC                     ;ADD IN 2X
4798  F96E  65 51          ADC     HOLD1
4799  F970  85 66          STA     MLTTMP
4800  F972  A5 67          LDA     MLTTMP+1
4801  F974  6D 9F 02       ADC     HOLD2
4802  F977  85 67          STA     MLTTMP+1
4803  F979  A6 57          LDX     DINDEX          ;NOW SHIFT MLTTMP LEFT DHLINE TIMES TO FINIS
4804  F97B  BC 81 FE       LDY     DHLINE,X        ;MULTIPLY
4805  F97E  88          CONVR1: DEY                 ;LOOP N TIMES
4806  F97F  30 07          BMI     CONVR2
4807  F981  06 66          ASL     MLTTMP
4808  F983  26 67          ROL     MLTTMP+1
4809  F985  4C 7E F9       JMP     CONVR1
4810  F988  BC A5 FE    CONVR2: LDY     DIV2TB,X    ;NOW DIVIDE HCRSR TO ACCOUNT FOR PARTIAL BYT
4811  F98B  A5 55          LDA     COLCRS
4812  F98D  A2 07          LDX     #7              ;* TRICKY *
4813  F98F  88           CONVR3: DEY
4814  F990  30 0A          BMI     CONVR4
4815  F992  CA             DEX
4816  F993  46 56          LSR     COLCRS+1
4817  F995  6A             ROR     A
4818  F996  6E A1 02       ROR     TMPLBT          ;SAVE LOW BITS FOR MASK
4819  F999  4C 8F F9       JMP     CONVR3
4820  F99C  C8           CONVR4: INY                 ;SO Y IS ZERO UPON RETURN FROM THIS ROUTINE
4821  F99D  18             CLC
4822  F99E  65 66          ADC     MLTTMP          ;ADD SHIFTED COLCRS TO MLTTMP
4823  F9A0  85 66          STA     MLTTMP
4824  F9A2  90 02          BCC     CONVR5
4825  F9A4  E6 67          INC     MLTTMP+1
4826  F9A6  38           CONVR5: SEC                 ;* TRICKY *
4827  F9A7  6E A1 02     CONVR6: ROR     TMPLBT      ;SLIDE A "1" UP AGAINST LOW BITS (CONTINUE T
4828  F9AA  18             CLC
4829  F9AB  CA             DEX                     ;AND FINISH SHIFT SO LOW BITS ARE
4830  F9AC  10 F9          BPL     CONVR6          ;RIGHT JUSTIFIED.
4831  F9AE  AE A1 02       LDX     TMPLBT          ;TMPLBT IS NOW THE INDEX INTO DMASKTB
4832  F9B1  A5 66          LDA     MLTTMP          ;PREPARE FOR RETURN
4833  F9B3  18             CLC
4834  F9B4  65 64          ADC     ADRESS
4835  F9B6  85 64          STA     ADRESS
4836  F9B8  85 5E          STA     OLDADR          ;REMEMBER THIS ADDRESS FOR CURSOR
4837  F9BA  A5 67          LDA     MLTTMP+1
4838  F9BC  65 65          ADC     ADRESS+1
```

```
4839  F9BE  85 65          STA     ADRESS+1
4840  F9C0  85 5F          STA     OLDADR+1
4841  F9C2  BD B1 FE       LDA     DMASKT,X
4842  F9C5  8D A0 02       STA     DMASK
4843  F9C8  85 6F          STA     SHFAMT
4844  F9CA  68             PLA
4845  F9CB  85 56          STA     COLCRS+1
4846  F9CD  68             PLA
4847  F9CE  85 55          STA     COLCRS
4848  F9D0  68             PLA
4849  F9D1  85 54          STA     ROWCRS
4850  F9D3  60             RTS
4851
4852
4853                       ; INCREMENT CURSOR AND DETECT BOTH END OF LINE AND END OF SCREEN
4854                       ;
4855  F9D4  A9 00          INCRSB: LDA     #0          ;NON-EXTEND ENTRY POINT
4856  F9D6  F0 02                  BEQ     INCRSC
4857  F9D8  A9 9B          INCRSR: LDA     #$9B        ;SPECIAL CASE ELIMINATOR
4858  F9DA  85 7D          INCRSC: STA     INSDAT
4859  F9DC  E6 63          INCRSA: INC     LOGCOL      ;(INSCHR ENTRY POINT)
4860  F9DE  E6 55                  INC     COLCRS
4861  F9E0  D0 02                  BNE     INCRS2      ;DO HIGH BYTE
4862  F9E2  E6 56                  INC     COLCRS+1
4863  F9E4  A5 55          INCRS2: LDA     COLCRS      ;TEST END OF LINE
4864  F9E6  A6 57                  LDX     DINDEX
4865  F9E8  DD 8D FE               CMP     COLUMN,X    ;TEST TABLED VALUE FOR ALL SCREEN MODES
4866  F9EB  F0 0B                  BEQ     INC2A       ;DO CR IF EQUAL
4867  F9ED  E0 00                  CPX     #0          ;MODE 0?
4868  F9EF  D0 06                  BNE     INCRS3      ;IF NOT, JUST RETURN
4869  F9F1  C5 53                  CMP     RMARGN      ;TEST AGAINST RMARGN
4870  F9F3  F0 02                  BEQ     INCRS3      ;EQUAL IS OK
4871  F9F5  B0 01                  BCS     INC2A       ;IF GREATER THAN, DO CR
4872  F9F7  60             INCRS3: RTS
4873  F9F8  E0 08          INC2A:  CPX     #8          ;CHECK MODE
4874  F9FA  90 04                  BCC     DOCR1       ;NOT 320X1 SO DO IT
4875  F9FC  A5 56                  LDA     COLCRS+1    ;TEST MSD
4876  F9FE  F0 F7                  BEQ     INCRS3      ;ONLY AT 64 SO DON'T DO IT
4877  FA00  A5 57          DOCR1:  LDA     DINDEX      ;DON'T MESS WITH LOGMAP IF NO MODE ZERO
4878  FA02  D0 30                  BNE     DOCR
4879  FA04  A5 63                  LDA     LOGCOL      ;TEST LINE OVERRUN
4880  FA06  C9 51                  CMP     #81
4881  FA08  90 0A                  BCC     DOCR1B      ;IF LESS THAN 81 IT IS DEFINITELY NOT LINE 3
4882  FA0A  A5 7D                  LDA     INSDAT
4883  FA0C  F0 26                  BEQ     DOCR        ;ONLY DO LOG LINE OVERFLOW IF INSDAT <>0
4884  FA0E  20 30 FA               JSR     DOCRWS      ;LOG LINE OVERFLOW IS SPECIAL CASE
4885  FA11  4C 77 FA               JMP     INCRS1      ;RETURN
4886  FA14  20 34 FA       DOCR1B: JSR     DOCR        ;GET IT OVER WITH
4887  FA17  A5 54                  LDA     ROWCRS
4888  FA19  18                     CLC                 ;TEST LOGICAL LINE BIT MAP
4889  FA1A  69 78                  ADC     #120
4890  FA1C  20 25 FB               JSR     BITGET
4891  FA1F  90 08                  BCC     DOCR1A      ;DON'T EXTEND IF OVERRUN IS INTO MIDDLE OF L
4892  FA21  A5 7D                  LDA     INSDAT      ;DON'T EXTEND IF INSDAT IS ZERO
```

```
4893  FA23  F0 04          BEQ     DOCR1A      ;(INSCHR SPECIAL CASE)
4894  FA25  18             CLC                 ;INSERT "0" INTO BIT MAP
4895  FA26  20 A5 F8       JSR     INSLIA
4896  FA29  4C DD FB DOCR1A: JMP    DOLCOL      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
4897  FA2C  A9 00    NOSCRL: LDA     #0          ;DOCR WITHOUT SCROLL
4898  FA2E  F0 02            BEQ     NOSCR1      ;(UNCONDITIONAL)
4899  FA30  A9 9B    DOCRWS: LDA     #$9B        ;DOCR WITH SCROLLING (NORMAL MODE)
4900  FA32  85 7D    NOSCR1: STA     INSDAT
4901  FA34  20 E4 FC DOCR:   JSR     COLCR       ;PLACE COLCRS AT LEFT EDGE
4902  FA37  A9 00            LDA     #0
4903  FA39  85 56            STA     COLCRS+1
4904  FA3B  E6 54            INC     ROWCRS
4905  FA3D  A6 57    DOCR2:  LDX     DINDEX
4906  FA3F  A0 18            LDY     #24         ;SET UP SCROLL LOOP COUNTER
4907  FA41  24 7B            BIT     SWPFLG
4908  FA43  10 05            BPL     DOCR2A      ;BRANCH IF NORMAL
4909  FA45  A0 04            LDY     #4
4910  FA47  98               TYA
4911  FA48  D0 03            BNE     DOCR2B      ;(UNCONDITIONAL)
4912  FA4A  BD 99 FE DOCR2A: LDA     NOROWS,X    ;GET NO OF ROWS
4913  FA4D  C5 54    DOCR2B: CMP     ROWCRS
4914  FA4F  D0 26            BNE     INCRS1
4915  FA51  8C 9D 02        STY     HOLD3
4916  FA54  8A               TXA                 ;DON'T SCROLL IF MODE <> 0
4917  FA55  D0 20            BNE     INCRS1
4918  FA57  A5 7D            LDA     INSDAT      ;OR IF INSDAT = 0
4919  FA59  F0 1C            BEQ     INCRS1
4920                ;        LDA     INSDAT      IF INSDAT <> $9B THEN ROLL IN A 0
4921  FA5B  C9 9B            CMP     #$9B        ;TO EXTEND BOTTOM LOGICAL LINE
4922  FA5D  38               SEC
4923  FA5E  F0 01            BEQ     DOCR4B
4924  FA60  18               CLC
4925  FA61  20 AC FB DOCR4B: JSR     SCROLL      ;LOOP BACK TO HERE IF >1 SCROLLS
4926  FA64  EE BB 02         INC     SCRFLG
4927  FA67  C6 6C            DEC     BUFSTR      ;ROWS MOVE UP SO BUFSTR SHOULD TOO
4928  FA69  CE 9D 02         DEC     HOLD3
4929  FA6C  AD B2 02         LDA     LOGMAP
4930  FA6F  38               SEC                 ;FOR PARTIAL LINES, ROLL IN A "1"
4931  FA70  10 EF            BPL     DOCR4B      ;AGAIN IF PARTIAL LOGICAL LINE
4932  FA72  AD 9D 02         LDA     HOLD3       ;PLACE CURSOR AT NEW LINE NEAR THE BOTTOM
4933  FA75  85 54            STA     ROWCRS
4934  FA77  4C DD FB INCRS1: JMP     DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4935                ;
4936                ;
4937                ; SUBEND: SUBTRACT ENDPT FROM ROWAC OR COLAC.  (X=0 OR 2)
4938                ;
4939  FA7A  38      SUBEND: SEC
4940  FA7B  B5 70           LDA     ROWAC,X
4941  FA7D  E5 74           SBC     ENDPT
4942  FA7F  95 70           STA     ROWAC,X
4943  FA81  B5 71           LDA     ROWAC+1,X
4944  FA83  E5 75           SBC     ENDPT+1
4945  FA85  95 71           STA     ROWAC+1,X
4946  FA87  60              RTS
```

```
 4947
 4948
 4949                    ;
 4950                    ; RANGE: DO CURSOR RANGE TEST.  IF ERROR, POP STACK TWICE AND JMP RETURN
 4951                    ;        (ERANGE IS EDITOR ENTRY POINT AND TEST IF EDITOR IS OPEN.
 4952                    ;         IF IT ISNT IT OPENS THE EDITOR AND CONTINUES)
 4953  FA88 AD BF 02     ;
 4954  FA8B C9 04    ERANGE: LDA     BOTSCR      ;IF BOTSCR=4
 4955  FA8D F0 07            CMP     #4
 4956  FA8F A5 57            BEQ     RANGE       ;THEN IT IS IN MIXED MODE AND OK
 4957  FA91 F0 03            LDA     DINDEX      ;IF MODE = 0
 4958  FA93 20 FC F3         BEQ     RANGE       ;THEN IT IS IN EDITOR MODE AND OK
 4959  FA96 A9 27            JSR     EOPEN       ;IF NOT, OPEN EDITOR
 4960  FA98 C5 53    RANGE:  LDA     #39         ;***** RANGE CHECK RMARGN ***** SET UP AC
 4961  FA9A B0 02            CMP     RMARGN      ;***** RANGE CHECK RMARGN ***** COMPARE
 4962  FA9C 85 53            BCS     RANGE3      ;***** RANGE CHECK RMARGN ***** BRANCH GE
 4963  FA9E A6 57            STA     RMARGN      ;***** RANGE CHECK RMARGN ***** BAD SO STORE
 4964  FAA0 BD 99 FE RANGE3: LDX     DINDEX
 4965  FAA3 C5 54            LDA     NOROWS,X    ;CHECK ROWS
 4966  FAA5 90 2A            CMP     ROWCRS
 4967  FAA7 F0 28            BCC     RNGERR      ;(ERROR IF TABLE.GE.ROWCRS)
 4968  FAA9 E0 08            BEQ     RNGERR
 4969  FAAB D0 0A            CPX     #8          ;CHECK FOR 320X1
 4970  FAAD A5 56            BNE     RANGE1      ;SPECIAL CASE IT
 4971  FAAF F0 13            LDA     COLCRS+1
 4972  FAB1 C9 01            BEQ     RNGOK       ;IF HIGH BYTE IS 0, COL IS OK
 4973  FAB3 D0 1C            CMP     #1
 4974  FAB5 F0 04            BNE     RNGERR      ;IF >1, BAD
 4975  FAB7 A5 56            BEQ     RANGE2      ;IF 1, GO CHECK LOW BYTE
 4976  FAB9 D0 16    RANGE1: LDA     COLCRS+1    ;FOR OTHERS, NON-ZERO HIGH BYTE IS BAD
 4977  FABB BD 8D FE         BNE     RNGERR
 4978  FABE C5 55    RANGE2: LDA     COLUMN,X    ;CHECK LOW BYTE
 4979  FAC0 90 0F            CMP     COLCRS
 4980  FAC2 F0 0D            BCC     RNGERR
 4981  FAC4 A9 01            BEQ     RNGERR
 4982  FAC6 85 4C    RNGOK:  LDA     #SUCCES     ;SET STATUS OK
 4983  FAC8 A9 80            STA     DSTAT
 4984  FACA A6 11            LDA     #BRKABT     ;PREPARE BREAK ABORT STATUS
 4985  FACC 85 11            LDX     BRKKEY      ;CHECK BREAK KEY FLAG
 4986  FACE F0 06            STA     BRKKEY      ;'CLEAR' BREAK
 4987  FAD0 60               BEQ     RNGER2      ;IF BREAK, QUIT IMMEDIATELY AND RETURN TO CI
 4988  FAD1 20 D6 F7         RTS
 4989  FAD4 A9 8D    RNGERR: JSR     HOME        ;ON RANGE ERROR, BRING CURSOR BACK
 4990  FAD6 85 4C            LDA     #CRSROR     ;SHOW CURSOR OVERRANGE ERROR
 4991  FAD8 68       RNGER2: STA     DSTAT
 4992  FAD9 68       RNGER1: PLA                 ;RESTORE STACK (THIS ROUTINE IS ALWAYS 1 LEV
 4993  FADA A5 7B            PLA                 ;AWAY FROM RETURN TO CIO)
 4994  FADC 10 03            LDA     SWPFLG      ;IF SWAPPED, SWAP BACK
 4995  FADE 20 B9 FC         BPL     RETUR3
 4996  FAE1 4C 34 F6         JSR     SWAPA       ;AND DONT DO RETUR1
 4997                 RETUR3: JMP     RETUR1      ;RETURN TO CIO
 4998                    ;
 4999                    ;
 5000                    ; OFFCRS: RESTORE OLD DATA UNDER CURSOR SO IT CAN BE MOVED
```

```
5001                            ;
5002  FAE4  A0 00              OFFCRS: LDY     #0
5003  FAE6  A5 5D                     LDA     OLDCHR
5004  FAE8  91 5E                     STA     (OLDADR),Y
5005  FAEA  60                        RTS
5006                            ;
5007                            ;
5008                            ;
5009                            ; BITMAP ROUTINES:
5010                            ;
5011                            ; BITCON: PUT MASK IN BITMSK AND INDEX IN X
5012                            ; BITPUT: PUT CARRY INTO BITMAP
5013                            ; BITROL: ROL CARRY INTO BOTTOM OF BITMAP (SCROLL)
5014                            ; BITSET: SET PROPER BIT
5015                            ; BITCLR: CLEAR PROPER BIT
5016                            ; BITGET: RETURN CARRY SET IF BIT IS THERE
5017                            ; LOGGET: DO BITGET FOR LOGMAP INSTEAD OF TABMAP
5018                            ;
5019  FAEB  48                  BITCON: PHA
5020  FAEC  29 07                       AND     #7
5021  FAEE  AA                          TAX                     ;GET MASK
5022  FAEF  BD B9 FE                    LDA     MASKTB,X
5023  FAF2  85 6E                       STA     BITMSK
5024  FAF4  68                          PLA                     ;PROCESS INDEX
5025  FAF5  4A                          LSR     A
5026  FAF6  4A                          LSR     A
5027  FAF7  4A                          LSR     A
5028  FAF8  AA                          TAX
5029  FAF9  60                          RTS
5030                            ;
5031                            ;
5032  FAFA  2E B4 02            BITROL: ROL     LOGMAP+2
5033  FAFD  2E B3 02                    ROL     LOGMAP+1
5034  FB00  2E B2 02                    ROL     LOGMAP
5035  FB03  60                          RTS
5036                            ;
5037                            ;
5038  FB04  90 0C               BITPUT: BCC     BITCLR          ;AND RETURN
5039                            ; OTHERWISE FALL THROUGH TO BITSET AND RETURN
5040  FB06  20 EB FA            BITSET: JSR     BITCON
5041  FB09  BD A3 02                    LDA     TABMAP,X
5042  FB0C  05 6E                       ORA     BITMSK
5043  FB0E  9D A3 02                    STA     TABMAP,X
5044  FB11  60                          RTS
5045                            ;
5046  FB12  20 EB FA            BITCLR: JSR     BITCON
5047  FB15  A5 6E                       LDA     BITMSK
5048  FB17  49 FF                       EOR     #$FF
5049  FB19  3D A3 02                    AND     TABMAP,X
5050  FB1C  9D A3 02                    STA     TABMAP,X
5051  FB1F  60                          RTS
5052                            ;
5053  FB20  A5 54               LOGGET: LDA     ROWCRS
5054  FB22  18                  LOGET:  CLC
```

```
5055  FB23  69 78          LO2GET: ADC     #120
5056  FB25  20 EB FA        BITGET: JSR     BITCON
5057  FB28  18                      CLC
5058  FB29  BD A3 02                LDA     TABMAP,X
5059  FB2C  25 6E                   AND     BITMSK
5060  FB2E  F0 01                   BEQ     BITGE1
5061  FB30  38                      SEC
5062  FB31  60              BITGE1: RTS
5063                        ;
5064                        ;
5065                        ;
5066                        ;
5067                        ; INATAC: INTERNAL(CHAR) TO ATASCII(ATACHR) CONVERSION
5068                        ;
5069  FB32  AD FA 02        INATAC: LDA     CHAR
5070  FB35  A4 57                   LDY     DINDEX      ;IF GRAPHICS MODES
5071  FB37  C0 03                   CPY     #3
5072  FB39  B0 0F                   BCS     INATA1      ;THEN DON'T CHANGE CHAR
5073  FB3B  2A                      ROL     A
5074  FB3C  2A                      ROL     A
5075  FB3D  2A                      ROL     A
5076  FB3E  2A                      ROL     A
5077  FB3F  29 03                   AND     #3
5078  FB41  AA                      TAX
5079  FB42  AD FA 02                LDA     CHAR
5080  FB45  29 9F                   AND     #$9F
5081  FB47  1D FA FE                ORA     INTATA,X
5082  FB4A  8D FB 02        INATA1: STA     ATACHR
5083  FB4D  60                      RTS
5084                        ;
5085                        ;
5086                        ;
5087                        ; MOVLIN: MOVE 40 BYTES AT FRMADR TO TOADR SAVING OLD TOADR
5088                        ;       DATA IN THE LINBUF.  THEN MAKE NEXT FRMADR
5089                        ;       BE AT LINBUF FOR NEXT TRANSFER & TOADR=TOADR+40
5090                        ;
5091  FB4E  A9 02           MOVLIN: LDA     #LINBUF/256 ;SET UP ADRESS=LINBUF=$247
5092  FB50  85 65                   STA     ADRESS+1
5093  FB52  A9 47                   LDA     #LINBUF.AND.$FF
5094  FB54  85 64                   STA     ADRESS
5095  FB56  A0 27                   LDY     #39
5096  FB58  B1 66           MOVLI1: LDA     (TOADR),Y   ;SAVE TO DATA
5097  FB5A  85 50                   STA     TMPCHR
5098  FB5C  B1 68                   LDA     (FRMADR),Y  ;STORE DATA
5099  FB5E  91 66                   STA     (TOADR),Y
5100  FB60  A5 50                   LDA     TMPCHR
5101  FB62  91 64                   STA     (ADRESS),Y
5102  FB64  88                      DEY
5103  FB65  10 F1                   BPL     MOVLI1
5104  FB67  A5 65                   LDA     ADRESS+1    ;SET UP FRMADR=LAST LINE
5105  FB69  85 69                   STA     FRMADR+1
5106  FB6B  A5 64                   LDA     ADRESS
5107  FB6D  85 68                   STA     FRMADR
5108  FB6F  18                      CLC                 ;ADD 40 TO TOADR
```

```
5109  FB70  A5 66          LDA     TOADR
5110  FB72  69 28          ADC     #40
5111  FB74  85 66          STA     TOADR
5112  FB76  90 02          BCC     MOVLI2
5113  FB78  E6 67          INC     TOADR+1
5114  FB7A  60      MOVLI2: RTS
5115                ;
5116                ;
5117                ;
5118                ; EXTEND: EXTEND BIT MAP FROM ROWCRS (EXTEND LOGICAL LINE
5119                ;
5120  FB7B  08      EXTEND: PHP             ;SAVE CARRY
5121  FB7C  A0 17           LDY     #23
5122  FB7E  98      EXTEN1: TYA
5123  FB7F  20 22 FB        JSR     LD1GET
5124  FB82  08              PHP
5125  FB83  98              TYA
5126  FB84  18              CLC
5127  FB85  69 79           ADC     #121
5128  FB87  28              PLP
5129  FB88  20 04 FB        JSR     BITPUT
5130  FB8B  88      EXTEN3: DEY
5131  FB8C  30 04           BMI     EXTEN4
5132  FB8E  C4 54           CPY     ROWCRS
5133  FB90  B0 EC           BCS     EXTEN1
5134  FB92  A5 54   EXTEN4: LDA     ROWCRS
5135  FB94  18              CLC
5136  FB95  69 78           ADC     #120
5137  FB97  28              PLP
5138  FB98  4C 04 FB        JMP     BITPUT  ;STORE NEW LINE'S BIT AND RETURN
5139                ;
5140                ;
5141                ;
5142                ; CLRLIN: CLEAR LINE CURSOR IS ON
5143                ;
5144  FB9B  A5 52   CLRLIN: LDA     LMARGN
5145  FB9D  85 55           STA     COLCRS
5146  FB9F  20 47 F9        JSR     CONVRT
5147  FBA2  A0 27           LDY     #39
5148  FBA4  A9 00           LDA     #0
5149  FBA6  91 64   CLRLI1: STA     (ADRESS),Y
5150  FBA8  88              DEY
5151  FBA9  10 FB           BPL     CLRLI1
5152  FBAB  60              RTS
5153                ;
5154                ;
5155                ;
5156                ;
5157                ; SCROLL: SCROLL SCREEN
5158                ;
5159  FBAC  20 FA FA SCROLL: JSR     BITROL  ;ROLL IN CARRY
5160  FBAF  A5 58           LDA     SAVMSC  ;SET UP WORKING REGISTERS
5161  FBB1  85 64           STA     ADRESS
5162  FBB3  A5 59           LDA     SAVMSC+1
```

```
5163  FBB5  85 65          STA     ADRESS+1
5164  FBB7  A0 28          LDY     #40             ;LOOP
5165  FBB9  B1 64          LDA     (ADRESS),Y
5166  FBBB  A6 6A          LDX     RAMTOP          ;TEST FOR LAST LINE
5167  FBBD  CA             DEX
5168  FBBE  E4 65          CPX     ADRESS+1
5169  FBC0  D0 08          BNE     SCROL2
5170  FBC2  A2 D7          LDX     #$D7
5171  FBC4  E4 64          CPX     ADRESS
5172  FBC6  B0 02          BCS     SCROL2
5173  FBC8  A9 00          LDA     #0              ;YES SO STORE ZERO DATA FOR THIS ENTIRE LINE
5174  FBCA  A0 00          LDY     #0
5175  FBCC  91 64          STA     (ADRESS),Y
5176  FBCE  E6 64          INC     ADRESS
5177  FBD0  D0 E5          BNE     SCROL1
5178  FBD2  E6 65          INC     ADRESS+1
5179  FBD4  A5 65          LDA     ADRESS+1
5180  FBD6  C5 6A          CMP     RAMTOP
5181  FBD8  D0 DD          BNE     SCROL1
5182  FBDA  4C DD FB       JMP     DOLCOL          ;AND RETURN
5183
5184
5185                       ; DOLCOL: DO LOGICAL COLUMN FROM BITMAP AND COLCRS
5186
5187  FBDD  A9 00          DOLCOL: LDA     #0              ;START WITH ZERO
5188  FBDF  85 63                  STA     LOGCOL
5189  FBE1  A5 54                  LDA     ROWCRS
5190  FBE3  85 51                  STA     HOLD1
5191  FBE5  A5 51          DOLC01: LDA     HOLD1           ;ADD IN ROW COMPONENT
5192  FBE7  20 22 FB               JSR     LO1GET
5193  FBEA  B0 0C                  BCS     DOLC02          ;FOUND BEGINNING OF LINE
5194  FBEC  A5 63                  LDA     LOGCOL          ;ADD 40 AND LOOK BACK ONE
5195  FBEE  18                     CLC
5196  FBEF  69 28                  ADC     #40
5197  FBF1  85 63                  STA     LOGCOL
5198  FBF3  C6 51                  DEC     HOLD1           ;UP ONE LINE
5199  FBF5  4C E5 FB               JMP     DOLC01
5200  FBF8  18             DOLC02: CLC                     ;ADD IN COLCRS
5201  FBF9  A5 63                  LDA     LOGCOL
5202  FBFB  65 55                  ADC     COLCRS
5203  FBFD  85 63                  STA     LOGCOL
5204  FBFF  60                     RTS
5205
5206
5207
5208                       ; DOBUFC: COMPUTE BUFFER COUNT AS THE NUMBER OF BYTES FROM
5209                       ;         BUFSTR TO END OF LOGICAL LINE WITH TRAILING SPACES REMOVED
5210
5211  FC00  20 9D FC       DOBUFC: JSR     PHACRS
5212  FC03  A5 63                  LDA     LOGCOL
5213  FC05  48                     PHA
5214  FC06  A5 6C                  LDA     BUFSTR          ;START
5215  FC08  85 54                  STA     ROWCRS
5216  FC0A  A5 6D                  LDA     BUFSTR+1
```

```
5217  FC0C  85 55          STA     COLCRS
5218  FC0E  A9 01          LDA     #1
5219  FC10  85 6B          STA     BUFCNT
5220  FC12  A2 17  DOBUF1: LDX     #23         ;NORMAL
5221  FC14  A5 7B          LDA     SWPFLG      ;IF SWAPPED, ROW 3 IS THE LAST LINE ON SCREE
5222  FC16  10 02          BPL     DOB1
5223  FC18  A2 03          LDX     #3
5224  FC1A  E4 54  DOB1:   CPX     ROWCRS      ;TEST IF CRSR IS AT LAST SCREEN POSITION
5225  FC1C  D0 0B          BNE     DOBU1A
5226  FC1E  A5 55          LDA     COLCRS
5227  FC20  C5 53          CMP     RMARGN
5228  FC22  D0 05          BNE     DOBU1A
5229  FC24  E6 6B          INC     BUFCNT      ;YES, SO FAKE INCRSR TO AVOID SCROLLING
5230  FC26  4C 39 FC       JMP     DOBUF2
5231  FC29  20 D4 F9 DOBU1A: JSR    INCRSB
5232  FC2C  E6 6B          INC     BUFCNT
5233  FC2E  A5 63          LDA     LOGCOL
5234  FC30  C5 52          CMP     LMARGN
5235  FC32  D0 DE          BNE     DOBUF1      ;NOT YET EOL
5236  FC34  C6 54          DEC     ROWCRS      ;BACK UP ONE INCRSR
5237  FC36  20 99 F7       JSR     CRSRLF
5238  FC39  20 A2 F5 DOBUF2: JSR    GETPLT      ;TEST CURRENT COLUMN FOR NON-ZERO DATA
5239  FC3C  D0 17          BNE     DOBUF4      ;QUIT IF NON-ZERO
5240  FC3E  C6 6B          DEC     BUFCNT      ;DECREMENT COUNTER
5241  FC40  A5 63          LDA     LOGCOL      ;BEGINNING OF LOGICAL LINE YET?
5242  FC42  C5 52          CMP     LMARGN
5243  FC44  F0 0F          BEQ     DOBUF4      ;YES, SO QUIT
5244  FC46  20 99 F7       JSR     CRSRLF      ;BACK UP CURSOR
5245  FC49  A5 55          LDA     COLCRS      ;IF LOGCOL=RMARGN, GO UP 1 ROW
5246  FC4B  C5 53          CMP     RMARGN
5247  FC4D  D0 02          BNE     DOBUF3
5248  FC4F  C6 54          DEC     ROWCRS
5249  FC51  A5 6B  DOBUF3: LDA     BUFCNT
5250  FC53  D0 E4          BNE     DOBUF2      ;LOOP UNLESS BUFCNT JUST WENT TO ZERO
5251  FC55  68     DOBUF4: PLA
5252  FC56  85 63          STA     LOGCOL
5253  FC58  20 A8 FC       JSR     PLACRS
5254  FC5B  60              RTS
5255
5256
5257
5258
5259                  ;  STRBEG: MOVE BUFSTR TO BEGINNING OF LOGICAL LINE.
5260
5261  FC5C  20 DD FB STRBEG: JSR    DOLCOL      ;USE DOLCOL TO POINT HOLD1 AT BOL
5262  FC5F  A5 51          LDA     HOLD1
5263  FC61  85 6C          STA     BUFSTR
5264  FC63  A5 52          LDA     LMARGN
5265  FC65  85 6D          STA     BUFSTR+1
5266  FC67  60             RTS
5267
5268
5269
5270
```

```
5271                            ;
5272                            ; DELTIM: TIME TO DELETE A LINE IF IT IS EMPTY AND AN EXTENSION
5273                            ;
5274 FC6B A5 63        DELTIA:  LDA     LOGCOL          ;IF LOGCOL<>LMARGN
5275 FC6A C5 52                 CMP     LMARGN          ;THEN DONT MOVE UP ONE
5276 FC6C D0 02                 BNE     DELTIB          ;LINE BEFORE TESTING DELTIM
5277 FC6E C6 54                 DEC     ROWCRS
5278 FC70 20 DD F8      DELTIB:  JSR     DOLCOL
5279 FC73 A5 63         DELTIM:  LDA     LOGCOL          ;TEST FOR EXTENSION
5280 FC75 C5 52                 CMP     LMARGN
5281 FC77 F0 13                 BEQ     DELTI3          ;NO
5282 FC79 20 47 F9              JSR     CONVRT
5283 FC7C A5 53                 LDA     RMARGN          ;SET UP COUNT
5284 FC7E 38                    SEC
5285 FC7F E5 52                 SBC     LMARGN
5286 FC81 A8                    TAY
5287 FC82 B1 64         DELTI1:  LDA     (ADRESS),Y
5288 FC84 D0 06                 BNE     DELTI3          ;FOUND A NON-0 SO QUIT AND RETURN
5289 FC86 88                    DEY
5290 FC87 10 F9                 BPL     DELTI1
5291 FC89 4C DB F8      DELTI2:  JMP     DELLIB          ;DELETE A LINE AND RETURN
5292 FC8C 60            DELTI3:  RTS
5293                            ;
5294                            ;
5295                            ;
5296                            ; TSTCTL: SEARCH CNTRLS TABLE TO SEE IF ATACHR IS A CNTL CHAR
5297                            ;
5298 FC8D A2 2D        TSTCTL:  LDX     #45             ;PREPARE TO SEARCH TABLE
5299 FC8F BD C6 FE     TSTCT1:  LDA     CNTRLS,X
5300 FC92 CD FB 02              CMP     ATACHR
5301 FC95 F0 05                 BEQ     TSTCT2
5302 FC97 CA                    DEX
5303 FC98 CA                    DEX
5304 FC99 CA                    DEX
5305 FC9A 10 F3                 BPL     TSTCT1
5306 FC9C 60            TSTCT2:  RTS
5307                            ;
5308                            ;
5309                            ;
5310                            ; PUSH ROWCRS,COLCRS AND COLCRS+1
5311                            ;
5312 FC9D A2 02         PHACRS:  LDX     #2
5313 FC9F B5 54         PHACR1:  LDA     ROWCRS,X
5314 FCA1 9D B8 02              STA     TMPROW,X
5315 FCA4 CA                    DEX
5316 FCA5 10 F8                 BPL     PHACR1
5317 FCA7 60                    RTS
5318                            ;
5319                            ;
5320                            ; PULL COLCRS+1,COLCRS AND ROWCRS
5321                            ;
5322 FCA8 A2 02         PLACRS:  LDX     #2
5323 FCAA BD B8 02      PLACR1:  LDA     TMPROW,X
5324 FCAD 95 54                 STA     ROWCRS,X
```

```
5325  FCAF  CA          DEX
5326  FCB0  10 F8       BPL     PLACR1
5327  FCB2  60          RTS
5328                    ;
5329                    ;
5330                    ;
5331                    ; SWAP: IF MIXED MODE, SWAP TEXT CURSORS WITH REGULAR CURSORS
5332                    ;
5333  FCB3  20 B9 FC    SWAP:   JSR     SWAPA       ;THIS ENTRY POINT DOES RETUR1
5334  FCB6  4C 34 F6            JMP     RETUR1
5335  FCB9  AD BF 02    SWAPA:  LDA     BOTSCR
5336  FCBC  C9 18               CMP     #24
5337  FCBE  F0 17               BEQ     SWAP3
5338  FCC0  A2 0B               LDX     #11
5339  FCC2  B5 54       SWAP1:  LDA     ROWCRS,X
5340  FCC4  48                  PHA
5341  FCC5  BD 90 02            LDA     TXTROW,X
5342  FCC8  95 54               STA     ROWCRS,X
5343  FCCA  68                  PLA
5344  FCCB  9D 90 02            STA     TXTROW,X
5345  FCCE  CA                  DEX
5346  FCCF  10 F1               BPL     SWAP1
5347  FCD1  A5 7B               LDA     SWPFLG
5348  FCD3  49 FF               EOR     #$FF
5349  FCD5  85 7B               STA     SWPFLG
5350  FCD7  60          SWAP3:  RTS
5351                    ;
5352                    ;
5353                    ; CLICK: MAKE CLICK THROUGH KEYBOARD SPEAKER
5354                    ;
5355  FCD8  A2 7F       CLICK:  LDX     #$7F
5356  FCDA  8E 1F D0    CLICK1: STX     CONSOL
5357  FCDD  8E 0A D4            STX     WSYNC
5358  FCE0  CA                  DEX
5359  FCE1  10 F7               BPL     CLICK1
5360  FCE3  60                  RTS
5361                    ;
5362                    ;
5363                    ; COLCR: PUTS EITHER 0 OR LMARGN INTO COLCRS BASED ON MODE AND SWPFLG
5364                    ;
5365  FCE4  A9 00       COLCR:  LDA     #0
5366  FCE6  A6 7B               LDX     SWPFLG
5367  FCE8  D0 04               BNE     COLCR1
5368  FCEA  A6 57               LDX     DINDEX
5369  FCEC  D0 02               BNE     COLCR2
5370  FCEE  A5 52       COLCR1: LDA     LMARGN
5371  FCF0  85 55       COLCR2: STA     COLCRS
5372  FCF2  60                  RTS
5373                    ;
5374                    ;
5375                    ; PUTMSC: PUT SAVMSC INTO ADRESS
5376                    ;
5377  FCF3  A5 58       PUTMSC: LDA     SAVMSC      ;SET UP ADDRESS
5378  FCF5  85 64               STA     ADRESS
```

```
5379  FCF7  A5 59          LDA     SAVMSC+1
5380  FCF9  85 65          STA     ADRESS+1
5381  FCFB  60             RTS
5382
```

```
                                        .PAGE
                                ;
                                ;
                                ; DRAW -- DRAW A LINE FROM OLDROW,OLDCOL TO NEWROW,NEWCOL
                                ; (THE AL MILLER METHOD FROM BASKETBALL)
FCFC  A2 00                     DRAW:   LDX     #0
FCFE  A5 22                             LDA     ICCOMZ      ;TEST COMMAND: $11=DRAW $12=FILL
FD00  C9 11                             CMP     #$11
FD02  F0 08                             BEQ     DRAWA
FD04  C9 12                             CMP     #$12        ;TEST FILL
FD06  F0 03                             BEQ     DRAWB       ;YES
FD08  A0 84                             LDY     #NVALID     ;NO, SO RETURN INVALID COMMAND
FD0A  60                                RTS
FD0B  E8                        DRAWB:  INX
FD0C  8E B7 02                  DRAWA:  STX     FILFLG
FD0F  A5 54                             LDA     ROWCRS      ;PUT CURSOR INTO NEWROW,NEWCOL
FD11  85 60                             STA     NEWROW
FD13  A5 55                             LDA     COLCRS
FD15  85 61                             STA     NEWCOL
FD17  A5 56                             LDA     COLCRS+1
FD19  85 62                             STA     NEWCOL+1
FD1B  A9 01                             LDA     #1
FD1D  85 79                             STA     ROWINC      ;SET UP INITIAL DIRECTIONS
FD1F  85 7A                             STA     COLINC
FD21  38                                SEC
FD22  A5 60                             LDA     NEWROW      ;DETERMINE DELTA ROW
FD24  E5 5A                             SBC     OLDROW
FD26  85 76                             STA     DELTAR
FD28  B0 0D                             BCS     DRAW1       ;DO DIRECTION AND ABSOLUTE VALUE
FD2A  A9 FF                             LDA     #$FF        ;BORROW WAS ATTEMPTED
FD2C  85 79                             STA     ROWINC      ;SET DIRECTION=DOWN
FD2E  A5 76                             LDA     DELTAR
FD30  49 FF                             EOR     #$FF        ;DELTAR = |DELTAR|
FD32  18                                CLC
FD33  69 01                             ADC     #1
FD35  85 76                             STA     DELTAR
FD37  38                        DRAW1:  SEC
FD38  A5 61                             LDA     NEWCOL      ;NOW DELTA COLUMN
FD3A  E5 5B                             SBC     OLDCOL
FD3C  85 77                             STA     DELTAC
FD3E  A5 62                             LDA     NEWCOL+1    ;TWO-BYTE QUANTITY
FD40  E5 5C                             SBC     OLDCOL+1
FD42  85 78                             STA     DELTAC+1
FD44  B0 16                             BCS     DRAW2       ;DIRECTION AND ABSOLUTE VALUE
FD46  A9 FF                             LDA     #$FF        ;BORROW WAS ATTEMPTED
FD48  85 7A                             STA     COLINC      ;SET DIRECTION = LEFT
FD4A  A5 77                             LDA     DELTAC
FD4C  49 FF                             EOR     #$FF        ;DELTAC = |DELTAC|
FD4E  85 77                             STA     DELTAC
FD50  A5 78                             LDA     DELTAC+1
FD52  49 FF                             EOR     #$FF
FD54  85 78                             STA     DELTAC+1
FD56  E6 77                             INC     DELTAC      ;ADD ONE FOR TWOS COMPLEMENT
FD58  D0 02                             BNE     DRAW2
```

```
5437  FD5A  E6 78          INC     DELTAC+1
5438  FD5C  A2 02  DRAW2:  LDX     #2          ;ZERO RAM FOR DRAW LOOP
5439  FD5E  A0 00          LDY     #0
5440  FD60  84 73          STY     COLAC+1
5441  FD62  98     DRAW3A: TYA
5442  FD63  95 70          STA     ROWAC,X
5443  FD65  B5 5A          LDA     OLDROW,X
5444  FD67  95 54          STA     ROWCRS,X
5445  FD69  CA             DEX
5446  FD6A  10 F6          BPL     DRAW3A
5447  FD6C  A5 77          LDA     DELTAC      ;FIND LARGER ONE (ROW OR COL)
5448                ;      STA     COUNTR      (PREPARE COUNTR AND ENDPT)
5449                ;      STA     ENDPT
5450  FD6E  E8             INX                 ;MAKE X 0
5451  FD6F  A8             TAY
5452  FD70  A5 78          LDA     DELTAC+1
5453  FD72  85 7F          STA     COUNTR+1
5454  FD74  85 75          STA     ENDPT+1
5455  FD76  D0 0B          BNE     DRAW3       ;AUTOMATICALLY LARGER IF MSD>0
5456  FD78  A5 77          LDA     DELTAC
5457  FD7A  C5 76          CMP     DELTAR      ;LOW COL >LOW ROW?
5458  FD7C  B0 05          BCS     DRAW3       ;YES
5459  FD7E  A5 76          LDA     DELTAR
5460  FD80  A2 02          LDX     #2
5461  FD82  A8             TAY
5462  FD83  98     DRAW3:  TYA                 ;PUT IN INITIAL CONDITIONS
5463  FD84  85 7E          STA     COUNTR
5464  FD86  85 74          STA     ENDPT
5465  FD88  48             PHA                 ;SAVE AC
5466  FD89  A5 75          LDA     ENDPT+1     ;PUT LSB OF HIGH BYTE
5467  FD8B  4A             LSR     A           ;INTO CARRY
5468  FD8C  68             PLA                 ;RESTORE AC
5469  FD8D  6A             ROR     A           ;ROR THE 9 BIT ACUMULATOR
5470  FD8E  95 70          STA     ROWAC,X
5471  FD90  A5 7E  DRAW4A: LDA     COUNTR      ;TEST ZERO
5472  FD92  05 7F          ORA     COUNTR+1
5473  FD94  D0 03          BNE     DRAW11      ;IF COUNTER IS ZERO, LEAVE DRAW
5474  FD96  4C 42 FE       JMP     DRAW10
5475  FD99  18     DRAW11: CLC                 ;ADD ROW TO ROWAC (PLOT LOOP)
5476  FD9A  A5 70          LDA     ROWAC
5477  FD9C  65 76          ADC     DELTAR
5478  FD9E  85 70          STA     ROWAC
5479  FDA0  90 02          BCC     DRAW5
5480  FDA2  E6 71          INC     ROWAC+1
5481  FDA4  A5 71  DRAW5:  LDA     ROWAC+1     ;COMPARE ROW TO ENDPOINT
5482  FDA6  C5 75          CMP     ENDPT+1     ;IF HIGH BYTE OF ROW IS .LT. HIGH
5483  FDA8  90 14          BCC     DRAW6       ;BYTE OF ENDPT, BLT TO COLUMN
5484  FDAA  D0 06          BNE     DRAW5A
5485  FDAC  A5 70          LDA     ROWAC
5486  FDAE  C5 74          CMP     ENDPT       ;LOW BYTE
5487  FDB0  90 0C          BCC     DRAW6       ;ALSO BLT
5488  FDB2  18     DRAW5A: CLC                 ;GE SO MOVE POINT
5489  FDB3  A5 54          LDA     ROWCRS
5490  FDB5  65 79          ADC     ROWINC
```

```
5491  FDB7  85 54          STA   ROWCRS
5492  FDB9  A2 00          LDX   #0          ;AND SUBTRACT ENDPT FROM ROWAC
5493  FDBB  20 7A FA       JSR   SUBEND
5494  FDBE  18       DRAW6: CLC               ;DO SAME FOR COLUMN (DOUBLE BYTE ADD)
5495  FDBF  A5 72          LDA   COLAC         ;ADD
5496  FDC1  65 77          ADC   DELTAC
5497  FDC3  85 72          STA   COLAC
5498  FDC5  A5 73          LDA   COLAC+1
5499  FDC7  65 78          ADC   DELTAC+1
5500  FDC9  85 73          STA   COLAC+1
5501  FDCB  C5 75          CMP   ENDPT+1       ;COMPARE HIGH BYTE
5502  FDCD  90 27          BCC   DRAW8
5503  FDCF  D0 06          BNE   DRAW6A
5504  FDD1  A5 72          LDA   COLAC         ;COMPARE LOW BYTE
5505  FDD3  C5 74          CMP   ENDPT
5506  FDD5  90 1F          BCC   DRAW8
5507  FDD7  24 7A DRAW6A:  BIT   COLINC        ;+ OR - ?
5508  FDD9  10 10          BPL   DRAW6B
5509  FDDB  C6 55          DEC   COLCRS        ;DO DOUBLE BYTE DECREMENT
5510  FDDD  A5 55          LDA   COLCRS
5511  FDDF  C9 FF          CMP   #$FF
5512  FDE1  D0 0E          BNE   DRAW7
5513  FDE3  A5 56          LDA   COLCRS+1
5514  FDE5  F0 0A          BEQ   DRAW7         ;DON'T DEC IF ZERO
5515  FDE7  C6 56          DEC   COLCRS+1
5516  FDE9  10 06          BPL   DRAW7         ;(UNCONDITIONAL)
5517  FDEB  E6 55 DRAW6B:  INC   COLCRS        ;DO DOUBLE BYTE INCREMENT
5518  FDED  D0 02          BNE   DRAW7
5519  FDEF  E6 56          INC   COLCRS+1
5520  FDF1  A2 02 DRAW7:   LDX   #2            ;AND SUBTRACT ENDPT FROM COLAC
5521  FDF3  20 7A FA       JSR   SUBEND
5522  FDF6  20 96 FA DRAW8: JSR   RANGE
5523  FDF9  20 E0 F5       JSR   OUTPLT        ;PLOT POINT
5524  FDFC  AD B7 02       LDA   FILFLG        ;TEST RIGHT FILL
5525  FDFF  F0 2F          BEQ   DRAW9
5526  FE01  20 9D FC       JSR   PHACRS
5527  FE04  AD FB 02       LDA   ATACHR
5528  FE07  8D BC 02       STA   HOLD4
5529  FE0A  A5 54 DRAW8A:  LDA   ROWCRS        ;SAVE ROW IN CASE OF CR
5530  FE0C  48             PHA
5531  FE0D  20 DC F9       JSR   INCRSA        ;POSITION CURSOR ONE PAST DOT
5532  FE10  68             PLA                 ;RESTORE ROWCRS
5533  FE11  85 54          STA   ROWCRS
5534  FE13  20 96 FA DRAW8C: JSR  RANGE
5535  FE16  20 A2 F5       JSR   GETPLT        ;GET DATA
5536  FE19  D0 0C          BNE   DRAW8B        ;STOP IF NON-ZERO DATA IS ENCOUNTERED
5537  FE1B  AD FD 02       LDA   FILDAT        ;FILL DATA
5538  FE1E  8D FB 02       STA   ATACHR
5539  FE21  20 E0 F5       JSR   OUTPLT        ;DRAW IT
5540  FE24  4C 0A FE       JMP   DRAW8A        ;LOOP
5541  FE27  AD BC 02 DRAW8B: LDA  HOLD4
5542  FE2A  8D FB 02       STA   ATACHR
5543  FE2D  20 A8 FC       JSR   PLACRS
5544  FE30  38       DRAW9: SEC                ;DO DOUBLE BYTE SUBTRACT
```

```
5545  FE31  A5 7E          LDA     COUNTR
5546  FE33  E9 01          SBC     #1
5547  FE35  85 7E          STA     COUNTR
5548  FE37  A5 7F          LDA     COUNTR+1
5549  FE39  E9 00          SBC     #0
5550  FE3B  85 7F          STA     COUNTR+1
5551  FE3D  30 03          BMI     DRAW10
5552  FE3F  4C 90 FD       JMP     DRAW4A
5553  FE42  4C 34 F6 DRAW10: JMP   RETUR1
```

```
5554                            .PAGE
5555                    ;
5556                    ;
5557                    ; TABLES
5558                    ;
5559                    ;
5560                    ; MEMORY ALLOCATION
5561                    ;
5562 FE45 18 10 0A 0A  ALOCAT: .BYTE   24,16,10,10,16,28,52,100,196,196,196,196
5563 FE49 10 1C 34 64
5564 FE4D C4 C4 C4 C4
5565                    ;
5566                    ;
5567                    ; NUMBER OF DISPLAY LIST ENTRIES
5568                    ;
5569 FE51 17 17 0B 17  NUMDLE: .BYTE   23,23,11,23,47,47,95,95,97,97,97,97
5570 FE55 2F 2F 5F 5F
5571 FE59 61 61 61 61
5572 FE5D 13 13 09 13  MXDMDE: .BYTE   19,19,9,19,39,39,79,79,65,65,65,65 ;(EXT OF NUMDLE)
5573 FE61 27 27 4F 4F
5574 FE65 41 41 41 41
5575                    ;
5576                    ;
5577                    ; ANTIC CODE FROM INTERNAL MODE CONVERSION TABLE
5578                    ;
5579                    ;   INTERNAL          ANTIC CODE            DESCRIPTION
5580                    ;       0                 2                40X2X8  CHARACTERS
5581                    ;       1                 6                20X5X8      ""
5582                    ;       2                 7                20X5X16     ""
5583                    ;       3                 8                40X4X8  GRAPHICS
5584                    ;       4                 9                80X2X4      ""
5585                    ;       5                 A                80X4X4      ""
5586                    ;       6                 B                160X2X2     ""
5587                    ;       7                 D                160X4X2     ""
5588                    ;       8                 F                320X2X1     ""
5589                    ;       9                 SAME AS 8 BUT GTIA 'LUM' MODE
5590                    ;       10                SAME AS 8 BUT GTIA 'COL/LUM REGISTER' MODE
5591                    ;       11                SAME AS 8 BUT GTIA 'COLOR' MODE
5592                    ;
5593 FE69 02 06 07 08  ANCONV: .BYTE   2,6,7,8,9,$A,$B,$D,$F,$F,$F,$F ;ZEROS FOR RANGE TEST IN
5594 FE6D 09 0A 0B 0D
5595 FE71 0F 0F 0F 0F
5596                    ;
5597                    ;
5598                    ; PAGE TABLE TELLS WHICH DISPLAY LISTS ARE IN DANGER OF
5599                    ; CROSSING A 256 BYTE PAGE BOUNDARY
5600                    ;
5601 FE75 00 00 00 00  PAGETB: .BYTE   0,0,0,0,0,0,0,1,1,1,1,1
5602 FE79 00 00 00 01
5603 FE7D 01 01 01 01
5604                    ;
5605                    ;
5606                    ; THIS IS THE NUMBER OF LEFT SHIFTS NEEDED TO MULTIPLY
5607                    ; COLCRS BY 10,20, OR 40.  (ROWCRS*10)/(2**DHLINE)
```

```
5608                            ;
5609  FE81  02 01 01 00        DHLINE: .BYTE   2,1,1,0,0,1,1,2,2,2,2,2
5610  FE85  00 01 01 02
5611  FE89  02 02 02 02
5612                            ;
5613                            ;
5614                            ; COLUMN: NUMBER OF COLUMNS
5615                            ;
5616  FE8D  28 14 14 28         COLUMN: .BYTE   40,20,20,40,80,80,160,160,64,80,80,80 ;MODE 8 IS SPECIAL
5617  FE91  50 50 A0 A0
5618  FE95  40 50 50 50
5619                            ;
5620                            ;
5621                            ;
5622                            ; NOROWS: NUMBER OF ROWS
5623                            ;
5624  FE99  18 18 0C 18         NOROWS: .BYTE   24,24,12,24,48,48,96,96,192,192,192,192
5625  FE9D  30 30 60 60
5626  FEA1  C0 C0 C0 C0
5627                            ;
5628                            ;
5629                            ;
5630                            ;
5631                            ; DIV2TB: HOW MANY RIGHT SHIFTS FOR HCRSR FOR PARTIAL BYTE MODES
5632                            ;
5633  FEA5  00 00 00 02         DIV2TB: .BYTE   0,0,0,2,3,2,3,2,3,1,1,1
5634  FEA9  03 02 03 02
5635  FEAD  03 01 01 01
5636                            ;
5637                            ;
5638                            ; DMASKT: DISPLAY MASK TABLE
5639                            ;
5640  FEB1  00 FF F0 0F         DMASKT  .BYTE   $00,$FF,$F0,$0F
5641  FEB5  C0 30 0C 03                 .BYTE   $C0,$30,$0C,$03
5642                            ;
5643                            ; MASKTB: BIT MASK.   (ALSO PART OF DMASKTB! DO NOT SEPARATE)
5644                            ;
5645  FEB9  80 40 20 10         MASKTB: .BYTE   $80,$40,$20,$10,$08,$04,$02,$01
5646  FEBD  08 04 02 01
5647                            ;
5648                            ;
5649                            ;
5650                            ;
5651  FEC1  28 CA 94 46         COLRTB: .BYTE   $28,$CA,$94,$46,$00
5652  FEC5  00
5653                            ;
5654                            ;
5655                            ;
5656                            ;
5657                            ; CNTRLS: CONTROL CODES AND THEIR DISPLACEMENTS INTO THE
5658                            ;         CONTROL CHARACTER PROCESSORS
5659                            ;
5660  FEC6  1B                  CNTRLS: .BYTE   $1B
5661  FEC7  79 F7                       .WORD   ESCAPE
```

```
5662  FEC9  1C                    .BYTE   $1C
5663  FECA  7F F7                 .WORD   CRSRUP
5664  FECC  1D                    .BYTE   $1D
5665  FECD  8C F7                 .WORD   CRSRDN
5666  FECF  1E                    .BYTE   $1E
5667  FED0  99 F7                 .WORD   CRSRLF
5668  FED2  1F                    .BYTE   $1F
5669  FED3  AA F7                 .WORD   CRSRRT
5670  FED5  7D                    .BYTE   $7D
5671  FED6  B9 F7                 .WORD   CLRSCR
5672  FED8  7E                    .BYTE   $7E
5673  FED9  E6 F7                 .WORD   BS
5674  FEDB  7F                    .BYTE   $7F
5675  FEDC  10 F8                 .WORD   TAB
5676  FEDE  9B                    .BYTE   $9B
5677  FEDF  30 FA                 .WORD   DOCRWS
5678  FEE1  9C                    .BYTE   $9C
5679  FEE2  D4 F8                 .WORD   DELLIN
5680  FEE4  9D                    .BYTE   $9D
5681  FEE5  A4 F8                 .WORD   INSLIN
5682  FEE7  9E                    .BYTE   $9E
5683  FEE8  32 F8                 .WORD   CLRTAB
5684  FEEA  9F                    .BYTE   $9F
5685  FEEB  2D F8                 .WORD   SETTAB
5686  FEED  FD                    .BYTE   $FD
5687  FEEE  0A F9                 .WORD   BELL
5688  FEF0  FE                    .BYTE   $FE
5689  FEF1  6D F8                 .WORD   DELCHR
5690  FEF3  FF                    .BYTE   $FF
5691  FEF4  37 F8                 .WORD   INSCHR
5692
5693
5694
5695
5696
5697                              ; ATAINT: ATASCI TO INTERNAL TABLE
5698                              ;
5699  FEF6  40 00 20 60           ATAINT: .BYTE   $40,$00,$20,$60
5700                              ;
5701                              ;
5702                              ; INTATA: INTERNAL TO ATASCI TABLE
5703                              ;
5704  FEFA  20 40 00 60           INTATA: .BYTE   $20,$40,$00,$60
5705                              ;
5706                              ;
5707                              ; ATASCI: ATASCII CONVERSION TABLE
5708                              ;
5709  FEFE  6C 6A 3B 80           ATASCI: .BYTE   $6C,$6A,$3B,$80,$80,$6B,$2B,$2A ;LOWER CASE
5710  FF02  80 6B 2B 2A
5711  FF06  6F 80 70 75                   .BYTE   $6F,$80,$70,$75,$9B,$69,$2D,$3D
5712  FF0A  9B 69 2D 3D
5713  FF0E
5714  FF0E  76 80 63 80                   .BYTE   $76,$80,$63,$80,$80,$62,$78,$7A
5715  FF12  80 62 78 7A
```

```
5716  FF16  34 80 33 36        BYTE  $34,$80,$33,$36,$1B,$35,$32,$31
5717  FF1A  1B 35 32 31
5718  FF1E
5719  FF1E  2C 20 2E 6E        BYTE  $2C,$20,$2E,$6E,$80,$6D,$2F,$81
5720  FF22  80 6D 2F 81
5721  FF26  72 80 65 79        BYTE  $72,$80,$65,$79,$7F,$74,$77,$71
5722  FF2A  7F 74 77 71
5723  FF2E
5724  FF2E  39 80 30 37        BYTE  $39,$80,$30,$37,$7E,$38,$3C,$3E
5725  FF32  7E 38 3C 3E
5726  FF36  66 68 64 80        BYTE  $66,$68,$64,$80,$82,$67,$73,$61
5727  FF3A  82 67 73 61
5728  FF3E
5729  FF3E
5730  FF3E  4C 4A 3A 80        BYTE  $4C,$4A,$3A,$80,$80,$4B,$5C,$5E ;UPPER CASE
5731  FF42  80 4B 5C 5E
5732  FF46  4F 80 50 55        BYTE  $4F,$80,$50,$55,$9B,$49,$5F,$7C
5733  FF4A  9B 49 5F 7C
5734  FF4E
5735  FF4E  56 80 43 80        BYTE  $56,$80,$43,$80,$80,$42,$58,$5A
5736  FF52  80 42 58 5A
5737  FF56  24 80 23 26        BYTE  $24,$80,$23,$26,$1B,$25,$22,$21
5738  FF5A  1B 25 22 21
5739  FF5E
5740  FF5E  5B 20 5D 4E        BYTE  $5B,$20,$5D,$4E,$80,$4D,$3F,$81
5741  FF62  80 4D 3F 81
5742  FF66  52 80 45 59        BYTE  $52,$80,$45,$59,$9F,$54,$57,$51
5743  FF6A  9F 54 57 51
5744  FF6E
5745  FF6E  28 80 29 27        BYTE  $28,$80,$29,$27,$9C,$40,$7D,$9D
5746  FF72  9C 40 7D 9D
5747  FF76  46 48 44 80        BYTE  $46,$48,$44,$80,$83,$47,$53,$41
5748  FF7A  83 47 53 41
5749  FF7E
5750  FF7E
5751  FF7E  0C 0A 7B 80        BYTE  $0C,$0A,$7B,$80,$80,$0B,$1E,$1F ;CONTROL
5752  FF82  80 0B 1E 1F
5753  FF86  0F 80 10 15        BYTE  $0F,$80,$10,$15,$9B,$09,$1C,$1D
5754  FF8A  9B 09 1C 1D
5755  FF8E
5756  FF8E  16 80 03 80        BYTE  $16,$80,$03,$80,$80,$02,$18,$1A
5757  FF92  80 02 18 1A
5758  FF96  80 80 85 80        BYTE  $80,$80,$85,$80,$1B,$80,$FD,$80
5759  FF9A  1B 80 FD 80
5760  FF9E
5761  FF9E  00 20 60 0E        BYTE  $00,$20,$60,$0E,$80,$0D,$80,$81
5762  FFA2  80 0D 80 81
5763  FFA6  12 80 05 19        BYTE  $12,$80,$05,$19,$9E,$14,$17,$11
5764  FFAA  9E 14 17 11
5765  FFAE
5766  FFAE  80 80 80 80        BYTE  $80,$80,$80,$80,$FE,$80,$7D,$FF
5767  FFB2  FE 80 7D FF
5768  FFB6  06 08 04 80        BYTE  $06,$08,$04,$80,$84,$07,$13,$01
5769  FFBA  84 07 13 01
```

```
5770                            ;
5771                            ;
5772                            ;
5773                            ;
5774                            ;
5775  FFBE  AD 09 D2    PIRQ5:  LDA     KBCODE
5776  FFC1  CD F2 02            CMP     CH1             ;TEST AGAINST LAST KEY PRESSED
5777  FFC4  D0 05               BNE     PIRQ3           ;IF NOT, GO PROCESS KEY
5778  FFC6  AD F1 02            LDA     KEYDEL          ;IF KEY DELAY BYTE > 0
5779  FFC9  D0 20               BNE     PIRQ4           ;IGNORE KEY AS BOUNCE
5780  FFCB  AD 09 D2    PIRQ3:  LDA     KBCODE          ;RESTORE AC
5781  FFCE  C9 9F              CMP     #CNTL1          ;TEST CONTROL 1 (SSFLAG)
5782  FFD0  D0 0A               BNE     PIRQ1
5783  FFD2  AD FF 02            LDA     SSFLAG
5784  FFD5  49 FF              EOR     #$FF
5785  FFD7  8D FF 02            STA     SSFLAG
5786  FFDA  B0 0F               BCS     PIRQ4           ;(UNCONDITIONAL) MAKE ^1 INVISIBLE
5787  FFDC  8D FC 02    PIRQ1:  STA     CH
5788  FFDF  8D F2 02            STA     CH1
5789  FFE2  A9 03               LDA     #3
5790  FFE4  8D F1 02            STA     KEYDEL          ;INITIALIZE KEY DELAY FOR DEBOUNCE
5791  FFE7  A9 00               LDA     #0              ;CLEAR COLOR SHIFT BYTE
5792  FFE9  85 4D               STA     ATRACT
5793  FFEB  A9 30       PIRQ4:  LDA     #$30
5794  FFED  8D 2B 02            STA     SRTIMR
5795  FFF0  68          PIRQ2:  PLA
5796  FFF1  40                  RTI
5797                            ;
5798                            ;
5799  FFF2  FF FF FF FF                 BYTE    $FF,$FF,$FF,$FF,$FF,$FF
5800  FFF6  FF FF
5801                            ;
5802  FFF8              CRNTPC  =*
5803                            *=$14
5804  0014  00          KBDSPR: .BYTE   $FFF8-CRNTPC ;^GDISPLC IS TOO LONG
5805  0015                      .END

ASSEMBLY ERRORS =    0
```

CROSS REFERENCE

LABEL	VALUE	REFERENCE
ACK	0041	-1632 1902
ACKREC	E9C6	1785 -1794
ADDCOR	030E	-498 2515 2519 2523
ADJ1	ED0C	2533 -2537
ADJUST	ED04	2470 2473 -2532
ADRESS	0064	-269 4120 4145 4171 4173 4182 4183 4184
		4186 4252 4254 4258 4260 4263 4265 4296
		4355 4357 4426 4443 4445 4570 4573 4574
		4653 4655 4681 4686 4748 4755 4758 4760
		4762 4766 4834 4835 4838 4839 5092 5094
		5101 5104 5106 5149 5161 5163 5165 5168
		5171 5175 5176 5178 5179 5287 5378 5380
ADRTAB	E6FE	-1292 1327
AFP	D800	-574
ALLPOT	D208	-659
ALLSEC	F30E	3886 -3891
ALOCAT	FE45	4146 -5562
ANCONV	FE69	4139 -5593
ANTIC	D400	-735 736 737 738 739 740 741 742
		743 744 745 746 747 748 749 750
APPEND	0001	-111
APPMHI	000E	-179 4761 4765
ASCC01	F705	4478 -4480
ASCZER	0030	-764 1193
ATACHR	02FB	-469 4308 4311 4316 4333 4372 4394 4417
		4421 4428 4481 4515 4526 4528 4606 5082
		5300 5527 5538 5542
ATAINT	FEF6	4343 -5699
ATAN	BE43	-602
ATASCI	FEFE	4480 -5709
ATEOF	F00B	3250 -3256
ATRACT	004D	-248 1340 1395 1401 1403 5792
AUDC1	D201	-667 2343 2353 2378
AUDC2	D203	-669 2354
AUDC3	D205	-671 2349
AUDC4	D207	-673
AUDCTL	D208	-674 2336
AUDF1	D200	-666 2297
AUDF2	D202	-668 2295
AUDF3	D204	-670 1749 2203 2607 3200
AUDF4	D206	-672 1751 2205 2609 3202
B192HI	0000	-1641 1750
B192LO	0028	-1640 1748
B600HI	0005	-1643 2204
B600LO	00CC	-1642 2202
BAD	EA63	1919 -1925
BADCOM	E9BF	1782 -1788 1809
BADDSK	F306	-3887 3923 3929
BADIOC	0086	-142 826

```
BADMOD    0091          -154  4141
BADST     EE9E          2914  2917  2963 -2965
BEEP      F058          3164  3197 -3314
BEEP1     F05A         -3315  3349
BEGIN     ED10          2255 -2558  2570
BELL      F90A          4402 -4726  5687
BELL1     F90C         -4727  4729
BFENHI    0035          -224  1774  1886  1985  2147  2184
BFENLO    0034          -223  1771  1883  1983  2145  2179
BITCLR    FB12          4624  5038 -5046
BITCON    FAEB         -5019  5040  5046  5056
BITGE1    FB31          5060 -5062
BITGET    FB25          4618  4890 -5056
BITMSK    006E          -276  5023  5042  5047  5059
BITPUT    FB04          4711 -5038  5129  5138
BITROL    FAFA         -5032  5159
BITSET    FB06          4622 -5040
BLACKB    F22A         -3740  3741
BLFILL    EEC1          2992 -2995  3015
BLIM      028A          -398  3183  3235  3254
BLKB2     F230          3740 -3742
BLKBDV    E471           -77  3483  3601  3603
BLOAD     F36C          3928 -3933
BOOT      F2CF          3710 -3862
BOOTAD    0242          -360  3896  3898  3934  3937
BOOT?     0009          -176  3864  3931  3991  4007
BOTSCR    02BF          -429  4193  4201  4542  4548  4664  4953  5335
BPTR      003D          -234  3182  3225  3234  3238  3246  3263  3265
                        3274  3293
BRKABT    0080          -136  2637  3188  4466  4983
BRKKEY    0011          -184  1337  1957  2064  2558  2580  2643  3189
                        3220  3831  4467  4509  4984  4985
BRKKY     0236          -345  1292  3833  3835
BRKKY2    E754         -1336  3832  3834
BROKE     EDA0          1959  2066  2560  2581 -2632
BS        F7E6         -4591  5673
BS1       F80D          4593 -4608
BS2       F805          4601  4603 -4605
BS3       F7F5          4596 -4598
BSA       F7EC         -4594
BUFADR    0015          -187  2829  2838  2842  2863  2865
BUFCNT    006B          -274  4386  4410  4412  5219  5229  5232  5240
                        5249
BUFFH     0004         -3421  3422  3883
BUFFL     0000         -3422  3881
BUFFUL    EECB          2989 -3000
BUFRFL    0038          -227  2054  2111  2160
BUFRHI    0033          -222  1773  1885  1980  1984  2142  2146  2182
                        2624  2626
BUFRLO    0032          -221  1769  1881  1951  1978  1982  2009  2133
                        2140  2144  2177  2615  2617  2621  2623
BUFSTR    006C          -275  4389  4391  4406  4408  4584  4588  4927
                        5214  5216  5263  5265
```

CAINI	F239	3669	-3750						
CART	BFFC	-3436	3658	3666	3781	3783	3784	3793	
CARTAD	BFFE	-3438	3750	3751	3788				
CARTCS	BFFA	-3435	3720	3726					
CARTFG	BFFD	-3437	3702	3705	3717	3723	3786		
CAS31	EC5E	2352	-2357						
CASBUF	03FD	-545	3101	3102	3237	3248	3253	3264	3295
		3301	3389	3391	3392	3421	3422	3892	3900
		3902	3905						
CASENT	EB80	1734	-2197						
CASET	0060	-1611	1732	2289	2351				
CASETV	E440	-59	3118	3513	3849				
CASFLG	030F	-499	1739	2050	2234				
CASINI	0002	-169	4009	4011	4014				
CASORG	EF41	-30	3097	3136					
CASRED	EBB3	2198	-2233						
CASSBT	004B	-245	3888	3922	3925	3975	4001	4005	
CASSET	0043	-124							
CASSPR	0014	-3398							
CAUX1	023C	-351	1763						
CAUX2	023D	-352	1765						
CBAUDH	02EF	-457	2602	2608	3144				
CBAUDL	02EE	-456	2601	2606	3142				
CBINI	F23C	3661	-3751						
CBUFH	0003	-3101	3102	3366					
CBUFHI	0002	-1669	1670	1772					
CBUFL	00FD	-3102	3368						
CBUFLO	003A	-1670	1768						
CCOMND	023B	-350	1760						
CDEVIC	023A	-349	1669	1670	1757				
CDTMA1	0226	-330	1528	2653	2655				
CDTMA2	0228	-331	1529						
CDTMF3	022A	-332	1455	3177	3179	3219	3222		
CDTMF4	022C	-334							
CDTMF5	022E	-336							
CDTMV1	0218	-323	1451	1452	1535	1537	1539	1540	1542
		1564	1566						
CDTMV2	021A	-324							
CDTMV3	021C	-325							
CDTMV4	021E	-326							
CDTMV5	0220	-327							
CDUBL	EF26	3072	-3075						
CH	02FC	-470	1478	4087	4462	4469	4474	5787	
CH1	02F2	-461	5776	5788					
CHACT	02F3	-463	1442	4110					
CHACTL	D401	-737	1443						
CHAR	02FA	-468	4302	4344	4346	5069	5079		
CHBAS	02F4	-464	1440	4108					
CHBASE	D409	-743	1441						
CHKDON	EABA	1995	-2002	2017					
CHKERR	008F	-151	2118						
CHKSNT	003B	-230	1947	1988	1994	2026			
CHKSUM	0031	-220	1946	1955	1991	2013	2015	2053	2115

```
                        2136  2138  2619
CHKTIM    EAF9        -2064  2071
CHRORG    E000          -22
CICLO2    E53F          901  -903
CICLOS    E533          853  -898
CIERR1    E4D1          821  -826
CIERR2    E6B0         1173 -1183
CIERR3    E50F         -869
CIERR4    E511          842  -870   874   880   920
CIJUMP    E693         1152 -1158
CINI      F3E1         3994 -4014
CIO       E4C4          771  -815
CIOCHR    002F         -217   815   951   956   966   985   989  1019
                       1026  1037  1083
CIOI1     E4A8         -787   798
CIOINT    E4A6          775  -786  1212
CIOINV    E46E          -76   774  3850
CIOORG    E4A6          -25   783
CIOPEN    E509          851  -864
CIOSPR    0014        -1215
CIOV      E456          -68   770  3683  3967
CIREAD    E569          857  -934
CIRT3     E62B        -1075  1080
CIRTN1    E61B          827   870   940  1009 -1065
CIRTN2    E61D          893   909   930   952   999  1060 -1069
CIST1     E559          916  -923
CISTSP    E54E          855  -914
CIWRIT    E5C9          858 -1003
CIX       00F2         -616
CKEY      004A         -244  3856  3997  4006
CKSTC     EE11         2808 -2810
CLICK     FCD8         4475  4727 -5355
CLICK1    FCDA        -5356  5359
CLOSE     000C          -93
CLOSEC    F02B         3119 -3286
CLRCHP    F27A        -3820  3825
CLRCOD    007D        -4030  4312
CLRLI1    FBA6        -5149  5151
CLRLIN    FB9B         4697 -5144
CLRRAM    F140        -3591  3594  3598
CLRSC2    F7BF        -4570  4572  4576
CLRSC3    F7CE        -4578  4581
CLRSCR    F7B9         4280  4314 -4567  5671
CLRTAB    F832        -4623  5683
CLRTBS    F430        -4125  4127
CLS       007D        -3416  3527
CLWRT     F038         3287 -3293
CMODE     EF1E        -3071  3087
CMPTAB    E6F6        -1282  1313
CNTL1     009F        -4031  5781
CNTRLS    FEC6         4442  4444  5299 -5660
COLAC     0072         -279  5440  5495  5497  5498  5500  5504
COLBK     D01A         -708
```

COLCR FCE4 4582 4901 -5365
COLCR1 FCEE 5367 -5370
COLCR2 FCF0 5369 -5371
COLCRS 0055 -259 4390 4409 4552 4553 4558 4560 4561
4587 4594 4599 4610 4679 4719 4778 4780
4811 4816 4845 4847 4860 4862 4863 4875
4903 4970 4975 4978 5145 5202 5217 5226
5245 5371 5400 5402 5509 5510 5513 5515
5517 5519
COLDST 0244 -361 3565 3606 3714
COLDSV E477 -79 3491
COLINC 007A -284 5406 5428 5507
COLON 003A -765
COLOR0 02C4 -436 4130
COLOR1 02C5 -437
COLOR2 02C6 -438
COLOR3 02C7 -439
COLOR4 02C8 -440 4169
COLPF0 D016 -704
COLPF1 D017 -705
COLPF2 D018 -706
COLPF3 D019 -707
COLPM0 D012 -700 1437
COLPM1 D013 -701
COLPM2 D014 -702
COLPM3 D015 -703
COLRSH 004F -250 1407 1435
COLRTB FEC1 4129 -5651
COLUMN FE8D 4865 4977 -5616
COM1 E647 1095 -1102
COM2 E662 1099 -1116
COMENT E63D 879 888 900 923 943 1012 -1093
COMFRM E978 -1748 1789
COMMND E974 -1743 1856
COMPLT 0043 -1634 1904
COMPUT ECA3 -2468 2600
COMRE1 F7A7 4556 -4559 4563 4564
COMRET F789 4541 -4545 4549
COMTAB E6C9 850 1107 -1209
CONSOL D01F -713 1432 3325 3330 3853 5356
CONTIN EC31 2307 -2331
CONVR1 F97E -4805 4809
CONVR2 F988 4806 -4810
CONVR3 F98F -4813 4819
CONVR4 F99C 4814 -4820
CONVR5 F9A6 4824 -4826
CONVR6 F9A7 -4827 4830
CONVRT F947 4295 4345 4652 4680 4720 -4776 5146 5282
COS BD73 -601
COUNT ED3B -2580 2591 2594 2604
COUNTR 007E -288 5453 5463 5471 5472 5545 5547 5548
5550
CR 009B -129 2991 3527 3539 3551 4317 4395 4420

		4510							
CRETRI	000D	-1672	1743	1802					
CRETRN	EBDF	2227	-2259						
CRETRY	0036	-225	1744	1788	1803				
CRITIC	0042	-239	1412	1729	1863				
CRLOOP	F5E7	-4329	4332						
CRNTP1	E6D5	-1213	1215						
CRNTP2	E90B	-1582	1584						
CRNTP3	EDE8	-2725	2727						
CRNTP4	EE78	-2872	2875						
CRNTP5	EF41	-3093	3097						
CRNTP6	F0E3	-3396	3398						
CRNTP7	F3E4	-4020	4023						
CRNTPC	FFF8	-5802	5804						
CRSINH	02F0	-459	1339	4122	4365				
CRSRDN	F78C	-4546	5665						
CRSRL1	F7A3	4554	-4557						
CRSRLF	F799	-4552	4598	5237	5244	5667			
CRSROR	008D	-149	4989						
CRSRRT	F7AA	-4560	4609	5669					
CRSRUP	F77F	-4540	4604	5663					
CSBOOT	F3B2	3694	-3989						
CSBOT2	F3C0	3990	-3997						
CSIDE	EF2E	3076	-3079						
CSIO	F0AC	-3370							
CSOPIV	E47D	-81	3132	4002					
CSTAT	0288	-396							
CTIA	D000	-681	682	683	684	685	686	687	688
		689	690	691	692	693	694	695	696
		697	698	699	700	701	702	703	704
		705	706	707	708	709	710	711	712
		713	714	715	716	717	718	719	720
		721	722	723	724	725	726	727	728
		729	730	731	732	733			
CTIMHI	0000	-1675	2446						
CTIMLO	0002	-1674	2445						
CTRLC	0092	-3417							
D	0044	-2901	3075						
DAUX1	030A	-494	1762	2960	3084	3880	3918		
DAUX2	030B	-495	1764	2210	2237	2259	3383	3878	
DBDDEC	F913	4188	-4740						
DBDEC	F91F	4178	-4751						
DBSECT	0241	-359	3916						
DBSUB	F921	4148	4165	4741	-4752				
DBSUB1	F934	4759	-4761						
DBUFHI	0305	-489	2181	2815	2864	3035	3367	3884	
DBUFLO	0304	-488	2176	2813	2862	3034	3369	3882	
DBUFSZ	0014	-2894	3077						
DBYTHI	0309	-493	2183	2820	2850	3049	3363		
DBYTLO	0308	-492	2178	2818	2848	3047	3365		
DCB	0300	-483							
DCOMND	0302	-486	1759	2799	2806	2824	2831	2959	3041
		3083	3361	3376	3872	3979			

```
DCTIM1    E8DD          1536 -1540
DCTIMR    E8DO          1409  1445  1454 -1535
DCTXF     E8EA          1538  1541  1543 -1546
DDEVIC    0300          -484  1731  1754  2288  2350  2797  3037  3204
                        3371
DECBF1    E66D          1124 -1126
DECBFL    E663           969   973  1040  1044 -1121
DEGFLG    00FB          -621
DEGON     0006          -624
DELAYO    EC8C         -2437  2441
DELAY1    EC8E         -2438  2439
DELCH1    F870         -4652  4669
DELCH2    F896          4662  4665 -4670
DELCHR    F86D         -4651  5689
DELETE    0021          -101
DELLI1    F8DD         -4703  4714
DELLI2    F8F8         -4718
DELLIA    F8D7         -4700
DELLIB    F8DB         -4702  4724  5291
DELLIN    F8D4         -4699  5679
DELTAC    0077          -282  5422  5425  5429  5431  5432  5434  5435
                        5437  5447  5452  5456  5496  5499
DELTAR    0076          -281  5410  5414  5418  5457  5459  5477
DELTI1    FC82         -5287  5290
DELTI2    FC89         -5291
DELTI3    FC8C          5281  5288 -5292
DELTIA    FC68          4673 -5274
DELTIB    FC70          5276 -5278
DELTIM    FC73          4597 -5279
DERR      E9F6          1821 -1842
DERR1     EA06          1791  1830 -1853
DERR5     F10D         -3539  3543  3544
DERRH     00F1         -3543  3544  3949
DERRL     000D         -3544  3948
DERROR    0090          -153  1910
DEVS1     E6A6         -1175  1180
DEVS2     E6B5          1176 -1188
DEVS3     E6C5          1195 -1197
DEVS4     E6C8          1185 -1201
DEVSRC    E69E           873   919 -1171
DFLAGS    0240          -358  3893
DHLINE    FE81          4804 -5609
DIGRT     00F1          -615
DINDEX    0057          -260  4106  4138  4194  4225  4363  4803  4864
                        4877  4905  4956  4963  5070  5368
DINI      F37E          3867  3930 -3941
DINIT     EDEA          2759 -2788
DIRECT    0002          -112
DISK      0044          -126
DISKID    0031         -2744  2796
DISKIV    E450           -66
DISKM     F3A4          3976 -3978
DISPLA    E410         -4056
```

```
DISPLY    0053          -122
DIV2TB    FEA5          4810 -5633
DLISTH    D403          -739  1424
DLISTL    D402          -738  1426
DMACTL    D400          -736  1428
DMASK     02A0          -416  4297  4351  4353  4842
DMASKT    FEB1          4841 -5640
DNACK     008B          -147  1914
DOBI      FC1A          5222 -5224
DOBOOT    F2ED          3874 -3877  3889
DOBU1A    FC29          5225  5228 -5231
DOBUF1    FC12         -5220  5235
DOBUF2    FC39          5230 -5238  5250
DOBUF3    FC51          5247 -5249
DOBUF4    FC55          5239  5243 -5251
DOBUFC    FC00          4405 -5211
DOCR      FA34          4613  4878  4883  4886 -4901
DOCR1     FA00          4874 -4877
DOCR1A    FA29          4891  4893 -4896
DOCR1B    FA14          4881 -4886
DOCR2     FA3D         -4905
DOCR2A    FA4A          4908 -4912
DOCR2B    FA4D          4911 -4913
DOCR4B    FA61          4923 -4925  4931
DOCRWS    FA30          4319  4419  4884 -4899  5677
DOINTP    ECE5         -2509  2512
DOLCD1    FBE5         -5191  5199
DOLCD2    FBF8          5193 -5200
DOLCOL    FBDD          4559  4608  4620  4648  4675  4698  4699  4725
                        4896  4934  5182 -5187  5261  5278
DOPEN     F3F6          4056 -4099
DOPEN1    F460         -4147  4150
DOPEN2    F4D5         -4203  4206
DOPEN3    F4FA         -4221  4224
DOPEN4    F51C         -4237  4240
DOPEN5    F524          4227 -4241
DOPEN7    F588          4279 -4284
DOPEN8    F438         -4129  4132
DOPEN9    F577          4271 -4277
DOPENA    F457          4140 -4143
DOSINI    000C          -178  3901  3903  3941  4008  4010
DOSS      F6AD          4397 -4431
DOSVEC    000A          -177  3602  3604  3730
DOUBLE    0044         -1626
DRAW      FCFC          4061 -5388
DRAW1     FD37          5411 -5419
DRAW10    FE42          5474  5551 -5553
DRAW11    FD99          5473 -5475
DRAW2     FD5C          5426  5436 -5438
DRAW3     FD83          5455  5458 -5462
DRAW3A    FD62         -5441  5446
DRAW4A    FD90         -5471  5552
DRAW5     FDA4          5479 -5481
```

```
DRAW5A    FDB2          5484 -5488
DRAW6     FDBE          5483  5487 -5494
DRAW6A    FDD7          5503 -5507
DRAW6B    FDEB          5508 -5517
DRAW7     FDF1          5512  5514  5516  5518 -5520
DRAW8     FDF6          5502  5506 -5522
DRAW8A    FE0A         -5529  5540
DRAW8B    FE27          5536 -5541
DRAW8C    FE13         -5534
DRAW9     FE30          5525 -5544
DRAWA     FD0C          5391 -5397
DRAWB     FD0B          5393 -5396
DRAWLN    0011           -98
DRETRI    0001         -1673  1741
DRETRY    0037          -226  1742  1853
DRKMSK    004E          -249  1406  1436
DSKFMS    001B          -191
DSKIF     EDF0          2760 -2796
DSKINV    E453           -67  3873  3982
DSKORG    EDEA           -28  2727  2769
DSKRDE    F381          3887  3921 -3948
DSKSPR    0014         -2875
DSKTIM    0246          -365  2789  2798  2830
DSKUTL    001A          -192
DSPFLG    02FE          -472  4437
DSTAT     004C          -246  4113  4142  4270  4369  4371  4393  4414
                        4423  4508  4745  4753  4769  4982  4990
DSTATS    0303          -487  1794  1826  1865  2197  2817  2851  3045
                        3381
DT1       00FA         -3110  3251  3296
DTA       00FC         -3109  3271
DTIMLO    0306          -490  2396  2803  3051  3375
DUNIT     0301          -485  1755  3039  3373  3870  3981
DUNUSE    0307          -491
DVSTAT    02EA          -455  2736  2737  2939  3059
EDITOR    E400         -4043
EDITRV    E400           -55  3515  3742  3744  3845  4039
EEXP      00ED          -611
EGETC1    F650         -4392  4403
EGETC2    F66E          4396 -4404
EGETC3    F67C          4387 -4410
EGETC5    F691          4411  4413 -4419
EGETC6    F66B          4401 -4403
EGETC7    F680         -4412  4415
EGETCH    F63E          4045 -4384
ENBOOT    F35A          3917 -3925
ENDACK    F1BD          3665  3667 -3674
ENDBCK    F1AC          3657  3659 -3663
ENDDIF    EE69          2833 -2851
ENDPT     0074          -280  4941  4944  5454  5464  5466  5482  5486
                        5501  5505
ENDRAM    F276          3802 -3810
ENINTP    ECEB         -2514
```

```
ENSPE2    F257          3782 -3794
ENSPEC    F254          3785  3787 -3793
ENTVEC    002C          -810
EOF       0088         -3418
EOFERR    0088          -144  3257  4507
EOL       009B          -766   967   990  1038  1055
EOPEN     F3FC          4043 -4102  4273  4958
EOT       00FE         -3111  3249  3304
EOUTC5    F6BE          4433 -4437
EOUTC6    F6B5         -4434  4439
EOUTCH    F6A4          4046 -4428
ERANGE    FA88          4385  4430 -4953
ERETN     F6BB         -4436
ERRFLG    023F          -356  1781  1818  1829  1842  1877  1922
ERROR     0045         -1635  1907
ERRTN     E4C0          -802   803   804
ERRTNH    00E4           791  -803   804   905
ERRTNL    00C0           789  -804   907
ESCAPE    F779         -4537  5661
ESCFLG    02A2          -418  4434  4438  4440  4538
ESIGN     00EF          -613
ESTSCM    F173          3607 -3622
EXP       DDC0          -595
EXP10     DDCC          -596
EXTEN1    FB7E         -5122  5133
EXTEN3    FB8B         -5130
EXTEN4    FB92          5131 -5134
EXTEND    FB7B          4677 -5120
FADD      DA66          -581
FASC      D8E6          -576
FCAX      F032         -3290  3298  3306
FCHRFL    00F0          -614
FDIV      DB2B          -583
FEOF      003F          -236  3162  3232  3256
FILDAT    02FD          -471  5537
FILFLG    02B7          -422  5397  5524
FILLBF    EEC3         -2996  2999
FILLIN    0012           -99
FINDX     ECD6         -2497
FLDOP     DD8D          -585
FLDOR     DD89          -584
FLD1P     DD9C          -587
FLD1R     DD98          -586
FLOPPY    0030         -1608
FLPTR     00FC          -625
FMOVE     DDB6          -590
FMSZPG    0043          -241
FMTD      EE4A         -2834
FMUL      DADB          -582
FNCNOT    0092          -155  1151
FOMAT     0021         -2749  2800  2832
FOOEY     EADE          2027 -2036
FORMAT    0022          -102
```

```
FPI         D9D2          -579
FPREC       0006          -572   634   635
FPSCR       05E6          -634   635   636
FPSCR1      05EC          -635   637
FPTR2       00FE          -626
FR0         00D4          -606
FR1         00E0          -608
FR2         00E6          -609
FRE         00DA          -607
FREQ        0040          -237  3314  3336
FRMADR      0068         -4033  4682  4687  5098  5105  5107
FRMERR      008C          -148  2102
FRX         00EC          -610
FSCR        05E6          -636
FSCR1       05EC          -637
FSTOP       DDAB          -589
FSTOR       DDA7          -588
FSUB        DA60          -580
FTYPE       003E          -235  3152  3382  4000
GBX         EFE8         -3240  3244
GBYTE       EFD6          3119 -3232  3255
GETCAR      0007         -3407
GETCH       F593          4058 -4290  4416
GETCHR      0007           -90
GETDAT      0040         -2751  2804
GETOUT      F749          4465 -4510
GETPLT      F5A2          4291 -4295  4361  4626  4639  4666  5238  5535
GETREC      0005           -89
GETSEC      F39D          3885  3919  3927 -3975
GLBABS      02E0          -447
GOBACK      EB09         -2072
GOERR       EF3D          3080 -3086
GOHAND      E689           883   902   924   950   955   984  1027  1056
                          -1151
GOOD        EA65          1903  1905  1923 -1927
GOODST      EE32          2822 -2824
GOON        EB64          2151 -2159
GOREAD      ED6F          2597 -2606
GPRIOR      026F          -369  1429  4151  4170
GRACTL      D01D          -711
GRAFM       D011          -699
GRAFP0      D00D          -695
GRAFP1      D00E          -696
GRAFP2      D00F          -697
GRAFP3      D010          -698
HARDI       F277          3583 -3818
HATABS      031A          -512   513  1102  1104  1175  3642
HDR         00FB         -3112
HITCLR      D01E          -712
HITIMR      ECCC         -2490  2494
HITONE      0005         -1644  2296
HOLD1       0051          -255  4143  4221  4237  4245  4700  4790  4798
                          5190  5191  5198  5262
```

```
HOLD2       029F          -415  4792  4801
HOLD3       029D          -413  4915  4928  4932
HOLD4       02BC          -426  5528  5541
HOLD5       02BD          -427
HOLDCH      007C          -286  4472  4512  4522
HOME        F7D6         -4582  4988
HOWMCH      F25F         -3798  3809
HPOSM0      D004          -686
HPOSM1      D005          -687
HPOSM2      D006          -688
HPOSM3      D007          -689
HPOSP0      D000          -682
HPOSP1      D001          -683
HPOSP2      D002          -684
HPOSP3      D003          -685
HSCROL      D404          -740
ICAX1       034A          -528  3682
ICAX1Z      002A          -213   935  1004  3153  4102  4104  4197  4277
                          4463
ICAX2       034B          -529
ICAX2Z      002B          -214  3070  3151  4099
ICBAH       0345          -523  1072  3680  3962
ICBAHZ      0025          -208  1073
ICBAL       0344          -522  1070  3678  3960
ICBALZ      0024          -207   959  1025  1071  1131  1133  1172  1192
ICBLH       0349          -527  1143
ICBLHZ      0029          -212  1145
ICBLL       0348          -526  1140  3966
ICBLLZ      0028          -211   947   948  1016  1017  1020  1121  1122
                          1125  1126  1141  1142  1144
ICCOM       0342          -520  3676  3964
ICCOMT      0017          -189   849   887  1106
ICCOMZ      0022          -205   840   934   961   979  1003  1032  1050
                          5389
ICDNO       0341          -519
ICDNOZ      0021          -204  1197
ICHID       0340          -518   788   928
ICHIDZ      0020          -203   864   904   914   929  1093  1189
ICIDNO      002E          -216   816   927  1069  1084  1138  1164
ICPTH       0347          -525   792
ICPTHZ      0027          -210   892   906
ICPTL       0346          -524   790
ICPTLZ      0026          -209   890   908
ICSPR       034C          -530
ICSPRZ      002C          -215   216   217   810   889   891  1103  1105
                          1109  1112  1113  1114  1159  1161
ICSTA       0343          -521
ICSTAZ      0023          -206   899   995  1065  1085  1153
IDENT       F0F2         -3527  3535  3536  3538
IDENTH      00F0         -3535  3536  3734
IDENTL      00F2         -3536  3733
IFP         D9AA          -577
IHINIT      E6D5          1230 -1269
```

```
IMASK     028B          -399
INATA1    FB4A          5072 -5082
INATAC    FB32          4292 -5069
INBUFF    00F3          -617
INC2A     F9F8          4866  4871 -4873
INCBF1    E676          1132 -1134
INCBFP    E670           960  1029 -1131
INCRS1    FA77          4885  4914  4917  4919 -4934
INCRS2    F9E4          4861 -4863
INCRS3    F9F7          4868  4870 -4872  4876
INCRSA    F9DC          4633 -4859  5531
INCRSB    F9D4          4293  4659 -4855  5231
INCRSC    F9DA          4856 -4858
INCRSR    F9D8          4322 -4857
INIMLH    0007         -3410  3843
INIMLL    0000         -3409  3841
INIT      EF41          3122 -3141
INSCH1    F852         -4637
INSCH3    F85E          4636 -4643
INSCH4    F844         -4630  4642
INSCH5    F86A          4645 -4648
INSCH6    F861         -4644  4647
INSCHR    F837         -4625  5691
INSCLR    0020          -117
INSDAT    007D          -287  4627  4637  4640  4858  4882  4892  4900
                        4918
INSLI1    F8C6         -4693  4696
INSLI2    F8CE          4692 -4697
INSLIA    F8A5         -4677  4895
INSLIN    F8A4         -4676  5681
INTABS    0200          -310  1232  1292  1293  1294  1295  1296  1297
                        1298  1299  1329  1331  1694  3631  4081
INTATA    FEFA          5081 -5704
INTEMP    022D          -335  1557  1563
INTINV    E46B           -75  1229  3852
INTORG    E6D5           -26  1215  1268
INTSPR    0014         -1584
INTTBL    EC84         -2419
INTZBS    0010          -182  3616
INVFLG    02B6          -421  4486  4488  4527
IOC1      E4D6           823  -830
IOC1A     E4D8          -831   836
IOC2      E4F3           847  -849
IOC6      E514           866  -873
IOC7      E519          -879
IOCB      0340          -517   831  1076
IOCBAS    0020          -202   832  1075
IOCBSZ    0010          -200   201   795
IOCFRE    00FF          -107   787   865   903   915
IRGEN     D20E          -678  1306  1308  1324  1326  2000  2034  2332
                        2374  4117
IRQST     D20E          -664  1302  1318
ISEOF     F00D          3233 -3257
```

```
ISRODN     EA90          1697 -1975  2420  2425  2426
ISRSIR     EB0F          1696 -2091  2419  2423  2424
ISRTD      EACF          1698 -2026  2421  2427  2428
JMPP       E735          1319 -1323
JSRIND     F6A1         -4426  4446
JTADRH     00EB         -2272  2273  2654
JTADRL     00EC         -2273  2652
JTIMER     EBEC         -2271  2272  2273
JTIMR1     E8CA          1411 -1528
JTIMR2     E8CD          1447 -1529
JVECK      028C          -400  1330  1332  1335  1354  1359  1362
K1         F729          4491 -4495
K2         F734          4496 -4500
K3         F73F          4501 -4505
K4         F776          4525 -4529
K5         F768          4514  4517  4519  4521 -4524
K6         F74D          4506 -4512
K7         F745          4468 -4508
K8         F773          4511 -4528
KBCODE     D209          -660  1477  5775  5780
KBD        004B          -121
KBDHND     E420         -4070
KBDORG     F3E4           -32  4023  4084
KBDSPR     0014         -5804
KEYBDV     E420           -57  3353  3355  3519  3847  4065
KEYDEL     02F1          -460  1463  1465  5778  5790
KGETC1     F71E          4485 -4490
KGETC2     F6DD         -4461  4483  4489  4494  4499  4504
KGETC3     F6FE         -4476  4523
KGETCH     F6E2          4072  4392 -4463  4471
LBFEND     05FF          -638
LBPR1      057E          -630
LBPR2      057F          -631
LBUFF      0580          -632   633
LDPNTR     EB6A          1805  1838 -2175  2223  2250
LEDGE      0002          -252  3622
LENGTH     022F         -1212
LFRTCM     F7A5         -4558  4566
LINBUF     0247          -367  5091  5093
LINZBS     0000          -166
LIRG       0000         -3419
LL         E72F          1317 -1320
LMARGN     0052          -256  3623  4282  4555  4565  4592  4595  4611
                         4678  4718  5144  5234  5242  5264  5275  5280
                         5285  5370
LO1GET     FB22         -5054  5123  5192
LO2GET     FB23          4705 -5055
LOCKFL     0023          -103
LOG        DECD          -597
LOG10      DED1          -598
LOGCOL     0063          -268  4399  4583  4591  4617  4621  4623  4631
                         4635  4657  4661  4859  4879  5188  5194  5197
                         5201  5203  5212  5233  5241  5252  5274  5279
```

```
LOGGET     FB20          4614  4722 -5053
LOGMAP     02B2          -420  4578  4715  4717  4929  5032  5033  5034
LOOPM      E71F         -1313  1321
LOOPM2     E72A          1315 -1318
LOTONE     0007         -1645  2294
LPENH      0234          -343  1422
LPENV      0235          -344  1420
MOPF       D000          -714
MOPL       D008          -722
M1PF       D001          -715
M1PL       D009          -723
M2PF       D002          -716
M2PL       D00A          -724
M3PF       D003          -717
M3PL       D00B          -725
MASKTB     FEB9          5022 -5645
MAXDEV     0021          -513  1094  1174
MAXIOC     0080          -201   797   822
MEMLO      02E7          -453  3842  3844
MEMORY   M 0000             0
MEMTOP     02E5          -452  3838  3840  4088  4259  4261
MLTTMP     0066          -270   271  4034  4784  4786  4787  4789  4791
                         4794  4796  4799  4800  4802  4807  4808  4822
                         4823  4825  4832  4837

MODATA     E9F0          1827 -1838
MODEM      004D          -125
MONORG     F0E3           -31  3398  3503
MONSPR     0014         -4023
MOTRGO     0034         -1664  2217  2244  3166  3206
MOTRST     003C         -1665  1706  2262  2633  3290
MOVLI1     FB58         -5096  5103
MOVLI2     FB7A          5112 -5114
MOVLIN     FB4E          4693 -5091
MOVVEC     F17D         -3630  3633
MVBUFF     F32D         -3904  3920
MVNXB      F32F         -3905  3908
MXDMDE     FE5D          4216 -5572
MXDMOD     0010          -116
N          004E         -2900  3071  3086
NACK       004E         -1633
NARG       0000             0
NBUFSZ     0028         -2893  3073
NCOMHI     003C         -1663  1709  2062
NCOMLO     0034         -1662  1776
NEWCOL     0061          -267  5401  5403  5420  5423
NEWROW     0060          -266  5399  5408
NLR        F005          3252 -3254
NMIEN      D40E          -748  1270
NMIRES     D40F          -749  1384
NMIST      D40F          -750  1372  1376
NOA1       F1F1          3701 -3703
NOA2       F212          3716  3719 -3721
NOB1       F1F8          3704 -3706
```

```
NOBOOT    F1FF          3707 -3713
NOCAR2    F220          3722 -3730
NOCART    F1FC          3699 -3710  3725
NOCKSM    003C          -231  1889  2150  2154
NOCLR     EAE9          2051 -2054
NOCSB2    F3BF          3993 -3995
NOCSBT    F3E0          3998 -4012
NODAT     0000         -2750
NOFUNC    F63D          4048  4073  4075 -4373
NOINIT    F2DC          3866 -3868
NOISE1    EC45          2341 -2343  2346
NOKEY     F2CE          3855 -3857
NOMOD     F4AB          4180 -4184
NONDEV    0082          -138  1183
NORMAL    004E         -1625
NOROWS    FE99          4912  4964 -5624
NOSCR1    FA32          4898 -4900
NOSCRL    FA2C         -4897
NOT8      F48B          4156  4167 -4170
NOTCAS    EC0C          2290 -2299
NOTCST    E96B          1733 -1738
NOTDER    EA52          1908 -1914
NOTDON    EA81         -1957  1962
NOTE      0026          -106
NOTEND    EA8E          1986 -2008
NOTERR    EA00          1843 -1849
NOTMXD    F4F5          4196  4199 -4219
NOTOPN    0085          -141   805  1098
NOTYET    EB3C          2112 -2131
NOWARM    F2DD          3863 -3869
NOWRPO    EA98          1979 -1982
NSIGN     00EE          -612
NTBRK0    EA88          1958 -1961
NTBRK1    EB00          2065 -2068
NTBRK2    ED17          2559 -2562
NTFRAM    EB1D          2100 -2105
NTOVRN    EB25          2106 -2111
NTWRP1    EB50          2141 -2144
NUMDLE    FE51          4216  4220 -5569
NVALID    0084          -140   839  5394
NWOK      EA56          1896  1912 -1917
NXTENT    F18C         -3641  3644
ODNHI     00EA         -2425  2426
ODNLO     0090         -2426
OFFCRS    FAE4          4404  4431 -5002
OKTIM1    ED1F          2565 -2568
OKTIMR    ED48          2584 -2588
OLDADR    005E          -265  4836  4840  5004
OLDCHR    005D          -264  4362  5003
OLDCOL    005B          -263  5421  5424
OLDROW    005A          -262  4330  5409  5443
OPEN      0003           -88   841
OPENC     EF4C          3119 -3151
```

```
OPINP     EF5D          3133  3156 -3160
OPNCOM    F404          4101 -4106
OPNEDT    F118         -3551  3553  3554
OPNERR    F453         -4141
OPNH      00F1         -3553  3554  3679
OPNIN     0004          -113   115
OPNIND    000C          -115
OPNL      0018         -3554  3677
OPNOT     0008          -114   115
OPNOUT    0002         -2892
OPNRTN    EF8F          3165 -3190  3198
OPNTMP    0066          -271  4153  4162  4191  4217  4218  4219
OPOK      EFD3          3184 -3226
OPOUT     EF95          3158 -3194
OPSYS     F17B         -3629
OSRAM     F28A          3634 -3831
OUTCH     F5B7          4059 -4308
OUTCH2    F5FF         -4344  4368  4630
OUTCHA    F5BD         -4311
OUTCHB    F5D7          4318 -4321
OUTCHE    F5CA          4313 -4316  4435
OUTPLT    F5E0          4321 -4326  4327  4607  5523  5539
OVRRUN    008E          -150  2108
POPF      D004          -718
POPL      D00C          -726
P1PF      D005          -719
P1PL      D00D          -727
P2PF      D006          -720
P2PL      D00E          -728
P3PF      D007          -721
P3PL      D00F          -729
PACTL     D302          -754  1272  1278  1345  1707  2218  2245  2263
                        2634  3167  3207  3291
PADDL0    0270          -371  1500
PADDL1    0271          -372
PADDL2    0272          -373
PADDL3    0273          -374
PADDL4    0274          -375  1502
PADDL5    0275          -376
PADDL6    0276          -377
PADDL7    0277          -378
PAGETB    FE75          4179 -5601
PALFLG    0000           -17  1647  1653  2483  2488  2538  2541  3168
                        3172  3209  3213  3317  3320  3340  3343
PBCTL     D303          -755  1273  1279  1349  1710  1777  2063  2635
PBPNT     001D          -195  2974  2984  2990  3001  3014
PBRK      EF8B         -3188  3221
PBUFSZ    001E          -196  2955  2988  2998  3046  3082
PBYTE     F010          3119 -3263
PCOLR0    02C0          -432  1434
PCOLR1    02C1          -433
PCOLR2    02C2          -434
PCOLR3    02C3          -435
```

PDEVN	0040	-2896	3036					
PENH	D40C	-746	1421					
PENV	D40D	-747	1419					
PHACR1	FC9F	-5313	5316					
PHACRS	FC9D	4625	4651	5211	-5312	5526		
PHCHLO	EE7F	-2940	3002	3003				
PHCLOS	EEDC	2913	-3013					
PHINIT	EE78	2918	-2932					
PHOPEN	EE9F	2912	-2972					
PHPUT	EF14	2964	-3059					
PHSTAT	EE81	2916	-2954	2972				
PHSTLO	EE7D	-2939	2956	2957				
PHWRIT	EEA7	2915	-2982					
PIA	D300	-751	752	753	754	755		
PIRQ	E6F3	-1281	1577	1578				
PIRQ1	FFDC	5782	-5787					
PIRQ2	FFF0	-5795						
PIRQ3	FFCB	5777	-5780					
PIRQ4	FFEB	5779	5786	-5793				
PIRQ5	FFBE	4082	-5775					
PIRQH	00E6	1252	-1577	1578				
PIRQL	00F3	1254	-1578					
PLACR1	FCAA	-5323	5326					
PLACRS	FCA8	4643	4674	5253	-5322	5543		
PLOT	0050	-1627						
PLUS	ECF8	2518	-2520					
PLYARG	05E0	-633	634					
PLYEVL	DD40	-591						
PMBASE	D407	-742						
PNMI	E791	-1372	1579	1580				
PNMI1	E799	1373	-1375					
PNMIH	00E7	1256	-1579	1580				
PNMIL	0091	1258	-1580					
POINT	0025	-105						
POKEY	D200	-650	651	652	653	654	655	656	657
		658	659	660	661	662	663	664	665
		666	667	668	669	670	671	672	673
		674	675	676	677	678	679		
POKMSK	0010	-183	1307	1316	1325	1997	1999	2031	2033
		2303	2329	2331	2372	2373	4115	4116	
POKTAB	EDD0	2521	2524	-2699					
PORTA	D300	-752	1275	1347	1482	1489			
PORTB	D301	-753	1276	1351					
POT0	D200	-651	1499						
POT1	D201	-652							
POT2	D202	-653							
POT3	D203	-654							
POT4	D204	-655	1501						
POT5	D205	-656							
POT6	D206	-657							
POT7	D207	-658							
POTGO	D208	-662	1505						
PRINTR	0050	-123							

```
PRINTV    E430          -58  3511  3848
PRIOR     D01B         -709  1430
PRMODE    EF1A         2983  3013 -3069
PRNBUF    03C0         -533  2940  2986  2996
PRNORG    EE78          -29  2875  2925
PRNSPR    0014        -3097
PRVOPN    0081         -137   869
PSIOC     EF01         3043 -3045
PTEMP     001F         -197  2982  2985
PTIMOT    001C         -194  2933  3050  3060
PTRIG0    027C         -383  1516
PTRIG1    027D         -384  1513
PTRIG2    027E         -385
PTRIG3    027F         -386
PTRIG4    0280         -387
PTRIG5    0281         -388
PTRIG6    0282         -389
PTRIG7    0283         -390
PTRLP     E8AC        -1509  1520
PUTADR    EE6D         2827  2834 -2862
PUTBC     EE43         2826 -2831
PUTCAR    000B        -3408
PUTCHR    000B          -92   886
PUTCNT    EE21         2811 -2817
PUTDAT    0080        -2752  2809
PUTDTO    EE01         2801 -2803
PUTLIN    F385         3735 -3958
PUTMSG    FCF3         4567  4782 -5377
PUTREC    0009          -91
PUTSEC    0050        -2745
PUTTXT    0009        -3406  3963
PWRONA    F3E4         4049  4062  4076 -4086
PWRUP     F125         3493  3497  3566 -3576  3686
PWRUP1    F128         3568 -3578
RADFLG    00FB         -622
RADON     0000         -623
RAMLO     0004         -170  3589  3590  3591  3595  3596  3795  3798
                       3800  3801  3804  3897  3899  3906  3910  3912
                       3913  3915  3936  3939  3940
RAMSIZ    02E4         -451  3655  3663  3837
RAMTOP    006A         -273  4090  4133  4144  4207  4229  4575  5166
                       5180
RANDOM    D20A         -661
RANGE     FA96         4290  4309  4955  4957 -4959  5522  5534
RANGE1    FAB7         4969 -4975
RANGE2    FABB         4974 -4977
RANGE3    FA9E         4961 -4963
RBLOK     EFE9         3130  3236 -3241
RBLOKV    E47A          -80  3129  3977
RCI1      E5A7          963   968  -973
RCI11     E5BF         -994
RCI1A     E574          936  -943
RCI1B     E571         -940   944
```

```
RCI2        E5AC          -979
RCI3        E587           949  -955   974
RCI4        E5C3           957   970   981   986  -998
RCI6        E5B2          -984   991
RDBAD       EE51         -2838  2845
RDBYTE      F310         -3892  3895
RDONLY      0087          -143  1008
READ        0052         -1616  3978
RECEIV      EAE0          1840  1891 -2048  2257
RECVDN      0039          -228  2055  2070  2122
RECVDS      EC60         -2371
RECVEN      EC1B          2061 -2320
REDGE       0027          -253  3624
RELONE      EAB1          1989 -1997
RENAME      0020          -100
RESET       F11B          3489  3499 -3564
RETUR1      F634          4044  4047  4057  4060  4070  4071  4074  4294
                          4364  4366 -4369  4529  4996  5334  5553
RETUR2      F621          4287  4315  4320  4323 -4361  4422  4447
RETUR3      FAE1          4994 -4996
RETURN      EA0D          1831  1851  1854 -1861  2265  2646
RIRGHI      0000         -1660  2241
RIRGLO      0078         -1655  2240
RMARGN      0053          -257  3625  4557  4562  4600  4869  4960  4962
                          5227  5246  5283
RNGER1      FAD8         -4991
RNGER2      FAD6          4986 -4990
RNGERR      FAD1          4966  4967  4973  4976  4979  4980 -4988
RNGOK       FAC4          4971 -4981
ROWAC       0070          -278  4940  4942  4943  4945  5442  5470  5476
                          5478  5480  5481  5485
ROWCRS      0054          -258  4329  4388  4407  4540  4544  4546  4547
                          4586  4602  4646  4663  4690  4701  4702  4776
                          4783  4849  4887  4904  4913  4933  4965  5053
                          5132  5134  5189  5215  5224  5236  5248  5277
                          5313  5324  5339  5342  5398  5444  5489  5491
                          5529  5533
ROWINC      0079          -283  5405  5413  5490
RRETRN      EB0E         -2081
RSIRQ       000A         -1657  2236
RTCLOK      0012          -185  1393  1396  1398  1404  2573  2599  3315
                          3334  3347
S           0053         -2902  3079
SAVADR      0068          -272  4033  4185  4187  4262  4264  4654  4656
                          4668  4672
SAVIO       0316          -505  2571  2590  2592
SAVMSC      0058          -261  4172  4174  4241  4243  5160  5162  5377
                          5379
SBUFSZ      001D         -2895  3081
SCOLLP      E80E         -1433  1439
SCREDT      0045          -120
SCRENV      E410           -56  3517  3846  4052
SCRFLG      02BB          -425  4629  4644  4926
```

SCRMEM	0093	-156	4768						
SCRNOK	F1DB	3685	-3687	3688	3690				
SCROL1	FBB7	4721	-5164	5177	5181				
SCROL2	FBCA	5169	5172	-5174					
SCROLL	FBAC	4925	-5159						
SDLSTH	0231	-339	1423						
SDLSTL	0230	-338	1425	4253	4255	4266	4268		
SDMCTL	022F	-337	1427	4111	4285	4286			
SECT1	F301	-3885	4003						
SECTX	F34C	-3919	3924						
SEND	EA6B	-1940	2225	2443					
SENDDS	EC5F	1861	1964	-2370	2632				
SENDEN	EBF2	1692	1943	2207	-2284				
SENDEV	E468	-74	1691	3205					
SENDIN	EC8A	1779	1807	-2436					
SERIN	D20D	-663	2114	2131					
SEROUT	D20D	-677	1952	1992	2010				
SETBSZ	EF34	3074	3078	-3082					
SETDCB	EEE6	2961	3004	-3034					
SETLOP	E8F7	-1561	1562						
SETTAB	F82D	-4621	5685						
SETVBL	E8ED	1224	1226	-1556					
SETVBV	E45C	-70	1225	2660	3178	3217			
SETVBX	EDB9	2215	2242	2253	2447	-2652			
SEX	0000	-3415	3675	3959					
SFH	EF64	-3163							
SHFAMT	006F	-277	4298	4347	4843				
SHFLOK	028E	-428	4092	4493	4498	4503	4520		
SHIFT1	F5B1	4299	-4302						
SHIFT2	F610	4348	-4351						
SHIFTD	F5AA	-4298	4301						
SHIFTU	F608	-4347	4350						
SIDWAY	0053	-1624							
SIGNON	F223	3485	-3733						
SIN	BD81	-600							
SIO	E959	1686	-1726						
SIOINT	E944	1689	-1706						
SIOINV	E465	-73	1688	3851					
SIOORG	E944	-27	1584	1702					
SIOSB	F095	3242	-3361	3394					
SIOSPR	0014	-2727							
SIOV	E459	-69	1685	2821	2962	3005	3384		
SIRHI	00EB	-2423	2424						
SIRLO	000F	-2424							
SIZEM	D00C	-694							
SIZEP0	D008	-690							
SIZEP1	D009	-691							
SIZEP2	D00A	-692							
SIZEP3	D00B	-693							
SKCTL	D20F	-679	1716	2300	2324				
SKRES	D20A	-676	2097	2326					
SKSTAT	D20F	-665	1459	1469	2096	2568	2588	2611	2613
SOUNDR	0041	-238	1715	2340					

```
SPACE      0020          -2899  2995
SPECIA     EF4B           3119 -3145
SPECIL     000E            -95   846   848
SPECL      F23F           3582 -3781
SQR        BEB1           -603
SRETRN     EB34           2116 -2121  2156
SRSTA      0040          -3103  3377
SRTIM2     0006          -1219  1475
SRTIMR     022B           -333  1467  1472  1476  1526  5794
SRTIR0     EB9B           2211 -2214
SRTIR1     EBC1           2238 -2241
SRTIR2     EBE9           2260 -2265
SSFLAG     02FF           -473  1338  4326  5783  5785
SSKCTL     0232           -340  1714  2285  2299  2321  2323  2612
STACK    S 0000              0
STACKP     0318           -507  1727  2640
STATC      0053          -2748  2810  2825  2958  3042  3871
STATIS     000D            -94
STATU      F028           3119 -3280
STATUS     0030           -219  1846  1849  1864  1894  1911  1915  1917
                          1927  1941  2060  2074  2103  2109  2119  2638
STATVH     0002          -2736  2737  2814
STATVL     00EA          -2737  2812
STICK0     0278           -379  1487  1491  1509
STICK1     0279           -380
STICK2     027A           -381
STICK3     027B           -382
STIMER     D209           -675
STLOOP     E877          -1482  1494
STORE      F917           4190  4204  4210  4212  4214  4222  4232  4234
                          4236  4238  4242  4244  4247  4249  4251  4257
                          4267  4269 -4745
STORE1     F91D          -4748
STRBEG     FC5C           4545 -5261
STRERR     F942           4764 -4768
STRIG0     0284           -391  1498
STRIG1     0285           -392
STRIG2     0286           -393
STRIG3     0287           -394
STRL       E890          -1497  1504
STROK      F946           4746  4754  4763  4767 -4770
STTMOT     EC75           1815  2252 -2396
SUBBFL     E677            998  1059 -1138
SUBEND     FA7A          -4939  5493  5521
SUBTMP     029E           -414  4752  4757
SUCCES     0001           -134   898  1850  1895  1940  2059  2993  3016
                          3226  3239  3266  3280  3289  4112  4370  4981
SUSUAL     EB38          -2124  2148  2162
SV7H       00E8          -1523  1524
SV7L       0073          -1524
SWAP       FCB3           4384  4398  4418  4424  4429  4436  4448 -5333
SWAP1      FCC2          -5339  5346
SWAP3      FCD7           5337 -5350
```

```
SWAPA     FCB9          4995  5333 -5335
SWPFLG    007B          -285  4121  4907  4993  5221  5347  5349  5366
SWSTA     0080         -3104  3380
SYIRQ     E706          1243 -1301
SYIRQ2    E71B          1304 -1310
SYIRQ8    E762          1322 -1343
SYIRQ9    E76F          1346 -1349
SYIRQA    E77A          1350 -1353
SYIRQB    E78F          1235  1236  1237  1240  1241  1242 -1364
SYRTI     E790          1234 -1365
SYRTI2    E78B          1358 -1362
SYSVB1    E7BA          1394  1397 -1399
SYSVB2    E7D6          1410 -1412
SYSVB3    E7E5          1417 -1419
SYSVB4    E832          1446 -1448
SYSVB6    E8C3          1471 -1525
SYSVB7    E873          1468  1473 -1480  1523  1524  1527
SYSVBA    E844          1453 -1456
SYSVBB    E834         -1449  1457
SYSVBL    E7AE          1227  1247 -1393
SYSVBV    E45F           -71
SYVB6A    E857          1461  1464 -1467
TAB       F810         -4609  4619  5675
TAB1      F823          4612  4615 -4617
TAB2      F82A          4616 -4620
TABMAP    02A3          -419  4125  5041  5043  5049  5050  5058
TBLENT    F0E3         -3510  3538  3641
TBLLEN    000E         -3538  3640
TDHI      00EA         -2427  2428
TDLO      00CF         -2428
TEMP      023E          -354  1667  1668  1901
TEMP1     0312          -502  2478  2496
TEMP2     0314          -503
TEMP3     0315          -504  2578  2596
TEMPHI    0002         -1667  1668  1884
TEMPLO    003E         -1668  1880
TIMER1    030C          -497  2472  2474  2477  2481  2574  2575
TIMER2    0310          -500  2468  2469  2471  2475  2479
TIMFLG    0317          -506  2068  2220  2247  2274  2564  2583  2662
TIMIT     EBA5         -2220  2221
TIMIT1    EBCB         -2247  2248
TIMOUT    008A          -146  1918  2073
TINDEX    0293          -409  4119
TMPCHR    0050          -254  4352  4356  5097  5100
TMPCOL    02B9          -424
TMPLBT    02A1          -417  4818  4827  4831
TMPROW    02B8          -423  5314  5323
TMPX1     029C          -412
TOADR     0066         -4034  4685  4689  5096  5099  5109  5111  5113
TONE1     0002         -3113  3196
TONE2     0001         -3114  3163
TOUT      EB0A          2069 -2073  2586
TOUT1     ED44          2566 -2585
```

```
TRAMSZ    0006          -171  3597  3654  3668  3697  3700  3715  3797
                        3805  3808  3836
TRIG0     D010          -730  1497
TRIG1     D011          -731
TRIG2     D012          -732
TRIG3     D013          -733
TRNRCD    0089          -145   994
TSTAT     0319          -508  1845  1928
TSTCT1    FC8F         -5299  5305
TSTCT2    FC9C          5301 -5306
TSTCTL    FC8D          4432  4524 -5298
TSTDAT    0007          -172  3653  3660  3698  3703  3721
TWICE     EE4F         -2836  2840
TXTCOL    0291          -408  4283
TXTMSC    0294          -410  4135  4137
TXTOLD    0296          -411
TXTROW    0290          -407  4281  5341  5344
UNLOCK    0024          -104
UPDNCM    F787         -4544  4551
USAREA    0480          -548
VBATRA    E7C8          1402 -1406
VBREAK    0206          -314  1361
VBWAIT    F496         -4175  4177
VCOUNT    D40B          -745  2572  2598  4175
VCTABL    E480           -24  1232  1694  3630  4081
VDELAY    D01C          -710
VDSLST    0200          -311  1232  1374
VECTBL    E400           -23
VIMIRQ    0216          -322  1281
VINTER    0204          -313  1352
VKEYBD    0208          -315  1293  4081
VPRCED    0202          -312  1348
VSCROL    D405          -741
VSERIN    020A          -316  1299  1309  1694
VSEROC    020E          -318  1297
VSEROR    020C          -317  1298
VTIMR1    0210          -319  1296
VTIMR2    0212          -320  1295
VTIMR4    0214          -321  1294
VVBLKD    0224          -329  1522
VVBLKI    0222          -328  1385
WAIT      EA1A         -1876  2449
WAITER    EC9B          1820 -2447
WAITTM    EF7C         -3179  3180
WARMST    0008          -175  3578  3584  3588  3862  3989
WARMSV    E474           -78  1379  3487
WATCOM    E9D7          1795 -1815
WCI1      E605          1034  1039 -1044
WCI1A     E5D4          1005 -1012
WCI1B     E5D1         -1009  1013
WCI2      E60A         -1050
WCI3      E5E5          1018 -1024  1045
WCI4      E5EB          1021 -1027
```

```
WCI5        E615          1028  1041  1052 -1059
WDLR        EFC6         -3220  3223
WFAK        F087          3337 -3350
WFAK1       F08C          3350 -3353
WFL         F060         -3324  3335
WIRGHI      0000         -1659  2214
WIRGLO      00B4         -1654  2213
WMODE       0289          -397  3161  3191  3195  3286
WOK         EA3D         -1901
WRITE       0057         -1617  2807
WRITEC      0057         -2898  3069
WRONLY      0083          -139   939
WSIOSB      F0D2          3272  3297  3305 -3389
WSIRG       000F         -1656  2209
WSYNC       D40A          -744  1560  5357
WTLR        F046          3294 -3299
XBOOT       F361          3926 -3928
XITVBL      E905          1228  1248  1418 -1571
XITVBV      E462           -72
XMTDON      003A          -229  1948  1961  2029
XXIT        E7E2          1413 -1418
ZERIT       EC6D         -2378  2381
ZERORM      F138         -3587
ZIOCB       0020          -199
ZOSRAM      F160          3585 -3610
ZOSRM2      F163         -3612  3615
ZOSRM3      F16E         -3617  3619
ZTBUF       F04A         -3301  3303
ZTEMP1      00F5          -618
ZTEMP3      00F9          -620
ZTEMP4      00F7          -619
```