ATARI®400/800™

OPERATING SYSTEM SOURCE LISTING

ATARI®

A Warner Communications Company

C017893

ATARI® 400/800™

OPERATING SYSTEM
SOURCE LISTING

NOTICE

TO ALL PERSONS RECEIVING THIS DOCUMENT

AUGUST 1981

```
                    LIST    X
THIS IS THE ORIGINAL JUNE 1979 ATARI 400/800™COMPUTER OPERATING
SYSTEM LISTING, MODIFIED TO ASSEMBLE ON THE MICROTEC CROSS
ASSEMBLER.  IN THE MODULES INTERUPT HANDLER AND MONITOR THERE IS
CODE WHICH IS USED FOR DEBUGGING.  IT ASSEMBLES TO ADDRESS $9000
TO $9020.  THIS IS NOT IN THE OS ROM AND THEREFORE NOT IN ANY
SYSTEMS EXCEPT THOSE USED FOR DEBUGGING.  IT IS INCLUDED FOR
HISTORICAL REASONS ONLY.
```

```
                        .PAGE
                ;
                ;
                ;       COLLEEN OPERATING SYSTEM EQUATE FILE
                ;
                ;
                ;
                ;       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       ;MOINTOR/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
```

```
 63 E453                DSKINV  =       $E453       ;DISK INTERFACE
 64 E456                CIOV    =       $E456       ;CENTRAL INPUT OUTPUT ROUTINE
 65 E459                SIOV    =       $E459       ;SERIAL INPUT OUTPUT ROUTINE
 66 E45C                SETVBV  =       $E45C       ;SET SYSTEM TIMERS ROUTINE
 67 E45F                SYSVBV  =       $E45F       ;SYSTEM VERTICAL BLANK CALCULATIONS
 68 E462                XITVBV  =       $E462       ;EXIT VERTICAL BLANK CALCULATIONS
 69 E465                SIOINV  =       $E465       ;SERIAL INPUT OUTPUT INITIALIZATION
 70 E468                SENDEV  =       $E468       ;SEND ENABLE ROUTINE
 71 E46B                INTINV  =       $E46B       ;INTERRUPT HANDLER INITIALIZATION
 72 E46E                CIOINV  =       $E46E       ;CENTRAL INPUT OUTPUT INITIALIZATION
 73 E471                BLKBDV  =       $E471       ;BLACKBOARD MODE
 74 E474                WARMSV  =       $E474       ;WARM START ENTRY POINT
 75 E477                COLDSV  =       $E477       ;COLD START ENTRY POINT
 76 E47A                RBLOKV  =       $E47A       ;CASSETTE READ BLOCK ENTRY POINT VECTOR
 77 E47D                CSOPIV  =       $E47D       ;CASSETTE OPEN FOR INPUT VECTOR
 78                     ;VCTABL =       $E480
 79                     ;
 80                     ;
 81                     ; OPERATING SYSTEM EQUATES
 82                     ;
 83                     ; COMMAND CODES FOR IOCB
 84 0003                OPEN    =       3           ;OPEN FOR INPUT/OUTPUT
 85 0005                GETREC  =       5           ;GET RECORD (TEXT)
 86 0007                GETCHR  =       7           ;GET CHARACTER(S)
 87 0009                PUTREC  =       9           ;PUT RECORD (TEXT)
 88 000B                PUTCHR  =       $B          ;PUT CHARACTER(S)
 89 000C                CLOSE   =       $C          ;CLOSE DEVICE
 90 000D                STATIS  =       $D          ;STATUS REQUEST
 91 000E                SPECIL  =       $E          ;BEGINNING OF SPECIAL ENTRY COMMANDS
 92                     ;
 93                     ; SPECIAL ENTRY COMMANDS
 94                     ;
 95 0011                DRAWLN  =       $11         ;DRAW LINE
 96 0012                FILLIN  =       $12         ;DRAW LINE WITH RIGHT FILL
 97 0020                RENAME  =       $20         ;RENAME DISK FILE
 98 0021                DELETE  =       $21         ;DELETE DISK FILE
 99 0022                FORMAT  =       $22         ;FORMAT
100 0023                LOCKFL  =       $23         ;LOCK FILE TO READ ONLY
101 0024                UNLOCK  =       $24         ;UNLOCK LOCKED FILE
102 0025                POINT   =       $25         ;POINT SECTOR
103 0026                NOTE    =       $26         ;NOTE SECTOR
104 00FF                IOCFRE  =       $FF         ;IOCB "FREE"
105                     ;
106                     ; AUX1 EQUATES
107                     ; () INDICATES WHICH DEVICES USE BIT
108                     ;
109 0001                APPEND  =       $1          ;OPEN FOR WRITE APPEND (D), OR SCREEN READ (
110 0002                DIRECT  =       $2          ;OPEN FOR DIRECTORY ACCESS (D)
111 0004                OPNIN   =       $4          ;OPEN FOR INPUT (ALL DEVICES)
112 0008                OPNOT   =       $8          ;OPEN FOR OUTPUT (ALL DEVICES)
113 000C                OPNINO  =       OPNIN+OPNOT ;OPEN FOR INPUT AND OUTPUT (ALL DEVICES)
114 0010                MXDMOD  =       $10         ;OPEN FOR MIXED MODE (E,S)
115 0020                INSCLR  =       $20         ;OPEN WITHOUT CLEARING SCREEN (E,S)
116                     ;
```

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

```
 171 0004                RAMLO:  .RES    2           ;RAM POINTER FOR MEMORY TEST
 172 0006                TRAMSZ: .RES    1           ;TEMPORARY REGISTER FOR RAM SIZE
 173 0007                TSTDAT: .RES    1           ;RAM TEST DATA REGISTER
 174                     ;
 175                     ; CLEARED ON COLDSTART ONLY
 176 0008                WARMST: .RES    1           ;WARM START FLAG
 177 0009                BOOT?:  .RES    1           ;SUCCESSFUL BOOT FLAG
 178 000A                DOSVEC: .RES    2           ;DISK SOFTWARE START VECTOR
 179 000C                DOSINI: .RES    2           ;DISK SOFTWARE INIT ADDRESS
 180 000E                APPMHI: .RES    2           ;APPLICATIONS MEMORY HI LIMIT
 181                     ;
 182                     ; CLEARED ON COLD OR WARM START
 183 0010                INTZBS  =*                  ;INTERRUPT HANDLER
 184 0010                POKMSK: .RES    1           ;SYSTEM MASK FOR POKEY IRQ ENABLE
 185 0011                BRKKEY: .RES    1           ;BREAK KEY FLAG
 186 0012                RTCLOK: .RES    3           ;REAL TIME CLOCK (IN 16 MSEC UNITS)
 187                     ;
 188 0015                BUFADR: .RES    2           ;INDIRECT BUFFER ADDRESS REGISTER
 189                     ;
 190 0017                ICCOMT: .RES    1           ;COMMAND FOR VECTOR
 191                     ;
 192 0018                DSKFMS: .RES    2           ;DISK FILE MANAGER POINTER
 193 001A                DSKUTL: .RES    2           ;DISK UTILITIES POINTER
 194                     ;
 195 001C                PTIMOT: .RES    1           ;PRINTER TIME OUT REGISTER
 196 001D                PBPNT:  .RES    1           ;PRINT BUFFER POINTER
 197 001E                PBUFSZ: .RES    1           ;PRINT BUFFER SIZE
 198 001F                PTEMP:  .RES    1           ;TEMPORARY REGISTER
 199                     ;
 200 0020                ZIOCB   =*                  ;ZERO PAGE I/O CONTROL BLOCK
 201 0010                IOCBSZ  =       16          ;NUMBER OF BYTES PER IOCB
 202 0080                MAXIOC  =       8*IOCBSZ    ;LENGTH OF THE IOCB AREA
 203 0020                IOCBAS  =*
 204 0020                ICHIDZ: .RES    1           ;HANDLER INDEX NUMBER (FF = IOCB FREE)
 205 0021                ICDNOZ: .RES    1           ;DEVICE NUMBER (DRIVE NUMBER)
 206 0022                ICCOMZ: .RES    1           ;COMMAND CODE
 207 0023                ICSTAZ: .RES    1           ;STATUS OF LAST IOCB ACTION
 208 0024                ICBALZ: .RES    1           ;BUFFER ADDRESS LOW BYTE
 209 0025                ICBAHZ: .RES    1
 210 0026                ICPTLZ: .RES    1           ;PUT BYTE ROUTINE ADDRESS - 1
 211 0027                ICPTHZ: .RES    1
 212 0028                ICBLLZ: .RES    1           ;BUFFER LENGTH LOW BYTE
 213 0029                ICBLHZ: .RES    1
 214 002A                ICAX1Z: .RES    1           ;AUXILLARY INFORMATION FIRST BYTE
 215 002B                ICAX2Z: .RES    1
 216 002C                ICSPRZ: .RES    4           ;TWO SPARE BYTES (CIO LOCAL USE)
 217 002E                ICIDNO  =       ICSPRZ+2    ;IOCB NUMBER X 16
 218 002F                CIOCHR  =       ICSPRZ+3    ;CHARACTER BYTE FOR CURRENT OPERATION
 219                     ;
 220 0030                STATUS: .RES    1           ;INTERNAL STATUS STORAGE
 221 0031                CHKSUM: .RES    1           ;CHECKSUM (SINGLE BYTE SUM WITH CARRY)
 222 0032                BUFRLO: .RES    1           ;POINTER TO DATA BUFFER (LO BYTE)
 223 0033                BUFRHI: .RES    1           ;POINTER TO DATA BUFFER (HI BYTE)
 224 0034                BFENLO: .RES    1           ;NEXT BYTE PAST END OF THE DATA BUFFER (LO B
```

```
    225 0035                BFENHI: .RES    1           ;NEXT BYTE PAST END OF THE DATA BUFFER (HI B
    226 0036                CRETRY: .RES    1           ;NUMBER OF COMMAND FRAME RETRIES
    227 0037                DRETRY: .RES    1           ;NUMBER OF DEVICE RETRIES
    228 0038                BUFRFL: .RES    1           ;DATA BUFFER FULL FLAG
    229 0039                RECVDN: .RES    1           ;RECEIVE DONE FLAG
    230 003A                XMTDON: .RES    1           ;TRANSMISSION DONE FLAG
    231 003B                CHKSNT: .RES    1           ;CHECKSUM SENT FLAG
    232 003C                NOCKSM: .RES    1           ;NO CHECKSUM FOLLOWS DATA FLAG
    233                     ;
    234                     ;
    235 003D                BPTR:   .RES    1
    236 003E                FTYPE:  .RES    1
    237 003F                FEOF:   .RES    1
    238 0040                FREQ:   .RES    1
    239 0041                SOUNDR: .RES    1           ;NOISY I/O FLAG.  (ZERO IS QUIET)
    240 0042                CRITIC: .RES    1           ;DEFINES CRITICAL SECTION (CRITICAL IF NON-Z
    241                     ;
    242 0043                FMSZPG: .RES    7           ;DISK FILE MANAGER SYSTEM ZERO PAGE
    243                     ;
    244                     ;
    245 004A                CKEY:   .RES    1           ;FLAG SET WHEN GAME START PRESSED
    246 004B                CASSBT: .RES    1           ;CASSETTE BOOT FLAG
    247 004C                DSTAT:  .RES    1           ;DISPLAY STATUS
    248                     ;
    249 004D                ATRACT: .RES    1           ;ATTRACT FLAG
    250 004E                DRKMSK: .RES    1           ;DARK ATTRACT MASK
    251 004F                COLRSH: .RES    1           ;ATTRACT COLOR SHIFTER (EOR'ED WITH PLAYFIEL
    252                     ;
    253 0002                LEDGE   =       2           ;LMARGN'S VALUE AT COLD START
    254 0027                REDGE   =       39          ;RMARGN'S VALUE AT COLD START
    255 0050                TMPCHR: .RES    1
    256 0051                HOLD1:  .RES    1
    257 0052                LMARGN: .RES    1           ;LEFT MARGIN (SET TO 1 AT POWER ON)
    258 0053                RMARGN: .RES    1           ;RIGHT MARGIN (SET TO 38 AT POWER ON)
    259 0054                ROWCRS: .RES    1           ;CURSOR COUNTERS
    260 0055                COLCRS  .RES    2
    261 0057                DINDEX: .RES    1
    262 0058                SAVMSC: .RES    2
    263 005A                OLDROW: .RES    1
    264 005B                OLDCOL: .RES    2
    265 005D                OLDCHR: .RES    1           ;DATA UNDER CURSOR
    266 005E                OLDADR: .RES    2
    267 0060                NEWROW: .RES    1           ;POINT DRAW GOES TO
    268 0061                NEWCOL: .RES    2
    269 0063                LOGCOL: .RES    1           ;POINTS AT COLUMN IN LOGICAL LINE
    270 0064                ADRESS: .RES    2
    271 0066                MLTTMP: .RES    2
    272 0066                OPNTMP  =       MLTTMP      ;FIRST BYTE IS USED IN OPEN AS TEMP
    273 0068                SAVADR: .RES    2
    274 006A                RAMTOP: .RES    1           ;RAM SIZE DEFINED BY POWER ON LOGIC
    275 006B                BUFCNT: .RES    1           ;BUFFER COUNT
    276 006C                BUFSTR: .RES    2           ;EDITOR GETCH POINTER
    277 006E                BITMSK: .RES    1           ;BIT MASK
    278 006F                SHFAMT: .RES    1
```

```
      279 0070                ROWAC:  .RES    2
      280 0072                COLAC:  .RES    2
      281 0074                ENDPT:  .RES    2
      282 0076                DELTAR: .RES    1
      283 0077                DELTAC: .RES    2
      284 0079                ROWINC: .RES    1
      285 007A                COLINC: .RES    1
      286 007B                SWPFLG: .RES    1       ;NON-0 IF TXT AND REGULAR RAM IS SWAPPED
      287 007C                HOLDCH: .RES    1       ;CH IS MOVED HERE IN KGETCH BEFORE CNTL & SH
      288 007D                INSDAT: .RES    1
      289 007E                COUNTR: .RES    2
      290                     ;
      291                     ;
      292                     ;
      293                     ;       80 - FF ARE RESERVED FOR USER APPLICATIONS
      294                     ;
      295                     ;
      296                     ;       NOTE : SEE FLOATING POINT SUBROUTINE AREA FOR ZERO PAGE CELLS
      297                     ;
      298                     ;
      299                     ;
      300                     ;       PAGE 1  -  STACK
      301                     ;
      302                     ;
      303                     ;
      304                     ;       PAGE TWO RAM ASSIGNMENTS
      305                     ;
      306                             *=$0200
      307 0200                INTABS  =*              ;INTERRUPT RAM
      308 0200                VDSLST: .RES    2       ;DISPLAY LIST NMI VECTOR
      309 0202                VPRCED: .RES    2       ;PROCEED LINE IRQ VECTOR
      310 0204                VINTER: .RES    2       ;INTERRUPT LINE IRQ VECTOR
      311 0206                VBREAK: .RES    2       ;SOFTWARE BREAK (00) INSTRUCTION IRQ VECTOR
      312 0208                VKEYBD: .RES    2       ;POKEY KEYBOARD IRQ VECTOR
      313 020A                VSERIN: .RES    2       ;POKEY SERIAL INPUT READY IRQ
      314 020C                VSEROR: .RES    2       ;POKEY SERIAL OUTPUT READY IRQ
      315 020E                VSEROC: .RES    2       ;POKEY SERIAL OUTPUT COMPLETE IRQ
      316 0210                VTIMR1: .RES    2       ;POKEY TIMER 1 IRQ
      317 0212                VTIMR2: .RES    2       ;POKEY TIMER 2 IRQ
      318 0214                VTIMR4: .RES    2       ;POKEY TIMER 4 IRQ
      319 0216                VIMIRQ: .RES    2       ;IMMEDIATE IRQ VECTOR
      320 0218                CDTMV1: .RES    2       ;COUNT DOWN TIMER 1
      321 021A                CDTMV2: .RES    2       ;COUNT DOWN TIMER2
      322 021C                CDTMV3: .RES    2       ;COUNT DOWN TIMER 3
      323 021E                CDTMV4: .RES    2       ;COUNT DOWN TIMER 4
      324 0220                CDTMV5: .RES    2       ;COUNT DOWN TIMER 5
      325 0222                VVBLKI: .RES    2       ;IMMEDIATE VERTICAL BLANK NMI VECTOR
      326 0224                VVBLKD: .RES    2       ;DEFERRED VERTICAL BLANK NMI VECTOR
      327 0226                CDTMA1: .RES    2       ;COUNT DOWN TIMER 1 JSR ADDRESS
      328 0228                CDTMA2: .RES    2       ;COUNT DOWN TIMER 2 JSR ADDRESS
```

```
    333 022A                CDTMF3: .RES    1           ;COUNT DOWN TIMER 3 FLAG
    334 022B                SRTIMR: .RES    1           ;SOFTWARE REPEAT TIMER
    335 022C                CDTMF4: .RES    1           ;COUNT DOWN TIMER 4 FLAG
    336 022D                INTEMP: .RES    1           ;IAN'S TEMP (RENAMED FROM T1 BY POPULAR DEMA
    337 022E                CDTMF5: .RES    1           ;COUNT DOWN TIMER FLAG 5
    338 022F                SDMCTL: .RES    1           ;SAVE DMACTL REGISTER
    339 0230                SDLSTL: .RES    1           ;SAVE DISPLAY LIST LOW BYTE
    340 0231                SDLSTH: .RES    1           ;SAVE DISPLAY LIST HI BYTE
    341 0232                SSKCTL: .RES    1           ;SKCTL REGISTER RAM
    342 0233                        .RES    1
    343                     ;
    344 0234                LPENH:  .RES    1           ;LIGHT PEN HORIZONTAL VALUE
    345 0235                LPENV:  .RES    1           ;LIGHT PEN VERTICAL VALUE
    346 0236                        .RES    2           ;SPARE BYTES
    347                     ;
    348 0238                        .RES    2           ;SPARE
    349                     ;
    350 023A                CDEVIC: .RES    1           ;COMMAND FRAME BUFFER - DEVICE
    351 023B                CCOMND: .RES    1           ;COMMAND
    352 023C                CAUX1:  .RES    1           ;COMMAND AUX BYTE 1
    353 023D                CAUX2:  .RES    1           ;COMMAND AUX BYTE 2
    354                     ;
    355                     ; NOTE: MAY NOT BE THE LAST WORD ON A PAGE
    356                     ;
    357 023E                TEMP:   .RES    1           ;TEMPORARY RAM CELL
    358                     ;
    359                     ; NOTE: MAY NOT BE THE LAST WORD ON A PAGE
    360                     ;
    361 023F                ERRFLG: .RES    1           ;ERROR FLAG - ANY DEVICE ERROR EXCEPT TIME O
    362                     ;
    363 0240                DFLAGS: .RES    1           ;DISK FLAGS FROM SECTOR ONE
    364 0241                DBSECT: .RES    1           ;NUMBER OF DISK BOOT SECTORS
    365 0242                BOOTAD: .RES    2           ;ADDRESS WHERE DISK BOOT LOADER WILL BE PUT
    366 0244                COLDST: .RES    1           ;COLDSTART FLAG (1=IN MIDDLE OF COLDSTART)
    367                     ;
    368 0245                        .RES    1           ;SPARE
    369                     ;
    370 0246                DSKTIM: .RES    1           ;DISK TIME OUT REGISTER
    371                     ;
    372 0247                LINBUF: .RES    40          ;CHAR LINE BUFFER
    373                     ;
    374 026F                GPRIOR: .RES    1           ;GLOBAL PRIORITY CELL
    375                     ;
    376 0270                PADDL0: .RES    1           ;POTENTIOMETER 0 RAM CELL
    377 0271                PADDL1: .RES    1
    378 0272                PADDL2: .RES    1
    379 0273                PADDL3: .RES    1
    380 0274                PADDL4: .RES    1
    381 0275                PADDL5: .RES    1
    382 0276                PADDL6: .RES    1
    383 0277                PADDL7: .RES    1
    384 0278                STICK0: .RES    1           ;JOYSTICK 0 RAM CELL
    385 0279                STICK1: .RES    1
    386 027A                STICK2: .RES    1
```

```
387  027B                STICK3: .RES  1
388  027C                PTRIG0: .RES  1          ;PADDLE TRIGGER 0
389  027D                PTRIG1: .RES  1
390  027E                PTRIG2: .RES  1
391  027F                PTRIG3: .RES  1
392  0280                PTRIG4: .RES  1
393  0281                PTRIG5: .RES  1
394  0282                PTRIG6: .RES  1
395  0283                PTRIG7: .RES  1
396  0284                STRIG0: .RES  1          ;JOYSTICK TRIGGER 0
397  0285                STRIG1: .RES  1
398  0286                STRIG2: .RES  1
399  0287                STRIG3: .RES  1
400                      ;
401  0288                CSTAT:  .RES  1
402  0289                WMODE:  .RES  1
403  028A                BLIM:   .RES  1
404                      ;
405  028B                        .RES  5          ;SPARE
406                      ;
407                      ;
408                      ;
409                      ;
410  0290                TXTROW: .RES  1          ;TEXT ROWCRS
411  0291                TXTCOL: .RES  2          ;TEXT COLCRS
412  0293                TINDEX: .RES  1          ;TEXT INDEX
413  0294                TXTMSC: .RES  2          ;FOOLS CONVRT INTO NEW MSC
414  0296                TXTOLD: .RES  6          ;OLDROW & OLDCOL FOR TEXT (AND THEN SOME)
415  029C                TMPX1:  .RES  1
416  029D                HOLD3:  .RES  1
417  029E                SUBTMP: .RES  1
418  029F                HOLD2:  .RES  1
419  02A0                DMASK:  .RES  1
420  02A1                TMPLBT: .RES  1
421  02A2                ESCFLG: .RES  1          ;ESCAPE FLAG
422  02A3                TABMAP: .RES  15
423  02B2                LOGMAP: .RES  4          ;LOGICAL LINE START BIT MAP
424  02B6                INVFLG: .RES  1          ;INVERSE VIDEO FLAG (TOGGLED BY ATARI KEY)
425  02B7                FILFLG: .RES  1          ;RIGHT FILL FLAG FOR DRAW
426  02B8                TMPROW: .RES  1
427  02B9                TMPCOL: .RES  2
428  02BB                SCRFLG: .RES  1          ;SET IF SCROLL OCCURS
429  02BC                HOLD4:  .RES  1          ;TEMP CELL USED IN DRAW ONLY
430  02BD                HOLD5:  .RES  1          ;DITTO
431  02BE                SHFLOK: .RES  1
432  02BF                BOTSCR: .RES  1          ;BOTTOM OF SCREEN : 24 NORM 4 SPLIT
433                      ;
434                      ;
435  02C0                PCOLR0: .RES  1          ;P0 COLOR
436  02C1                PCOLR1: .RES  1          ;P1 COLOR
437  02C2                PCOLR2: .RES  1          ;P2 COLOR
438  02C3                PCOLR3: .RES  1          ;P3 COLOR
439  02C4                COLOR0: .RES  1          ;COLOR 0
440  02C5                COLOR1: .RES  1
```

```
      441  02C6                COLOR2: .RES    1
      442  02C7                COLOR3: .RES    1
      443  02C8                COLOR4: .RES    1
      444                      ;
      445                      ;
      446  02C9                        .RES    23          ;SPARE
      447                      ;
      448                      ;
      449                      ;
      450  02E0                GLBABS  =*                  ;GLOBAL VARIABLES
      451                      ;
      452  02E0                        .RES    4           ;SPARE
      453                      ;
      454  02E4                RAMSIZ: .RES    1           ;RAM SIZE (HI BYTE ONLY)
      455  02E5                MEMTOP: .RES    2           ;TOP OF AVAILABLE USER MEMORY
      456  02E7                MEMLO:  .RES    2           ;BOTTOM OF AVAILABLE USER MEMORY
      457  02E9                        .RES    1           ;SPARE
      458  02EA                DVSTAT: .RES    4           ;STATUS BUFFER
      459  02EE                CBAUDL: .RES    1           ;CASSETTE BAUD RATE LOW BYTE
      460  02EF                CBAUDH: .RES    1
      461                      ;
      462  02F0                CRSINH: .RES    1           ;CURSOR INHIBIT (00 = CURSOR ON)
      463  02F1                KEYDEL: .RES    1           ;KEY DELAY
      464  02F2                CH1:    .RES    1
      465                      ;
      466  02F3                CHACT:  .RES    1           ;CHACTL REGISTER RAM
      467  02F4                CHBAS:  .RES    1           ;CHBAS REGISTER RAM
      468                      ;
      469  02F5                        .RES    5           ;SPARE BYTES
      470                      ;
      471  02FA                CHAR:   .RES    1
      472  02FB                ATACHR: .RES    1           ;ATASCII CHARACTER
      473  02FC                CH:     .RES    1           ;GLOBAL VARIABLE FOR KEYBOARD
      474  02FD                FILDAT: .RES    1           ;RIGHT FILL DATA (DRAW)
      475  02FE                DSPFLG: .RES    1           ;DISPLAY FLAG : DISPLAY CNTLS IF NON-ZERO
      476  02FF                SSFLAG: .RES    1           ;START/STOP FLAG FOR PAGING (CNTL 1). CLEARE
      477                      ;
      478                      ;
      479                      ;
      480                      ;
      481                      ;
      482                      ;
      483                      ;
      484                      ;       PAGE THREE RAM ASSIGNMENTS
      485                      ;
      486  0300                DCB     =*                  ;DEVICE CONTROL BLOCK
      487  0300                DDEVIC: .RES    1           ;PERIPHERAL UNIT 1 BUS I.D. NUMBER
      488  0301                DUNIT:  .RES    1           ;UNIT NUMBER
      489  0302                DCOMND: .RES    1           ;BUS COMMAND
      490  0303                DSTATS: .RES    1           ;COMMAND TYPE/STATUS RETURN
      491  0304                DBUFLO: .RES    1           ;DATA BUFFER POINTER LOW BYTE
      492  0305                DBUFHI: .RES    1
      493  0306                DTIMLO: .RES    1           ;DEVICE TIME OUT IN 1 SECOND UNITS
      494  0307                DUNUSE: .RES    1           ;UNUSED BYTE
```

```
 495 0308                DBYTLO: .RES    1           ;NUMBER OF BYTES TO BE TRANSFERRED LOW BYTE
 496 0309                DBYTHI: .RES    1
 497 030A                DAUX1:  .RES    1           ;COMMAND AUXILLARY BYTE 1
 498 030B                DAUX2:  .RES    1
 499                     ;
 500 030C                TIMER1: .RES    2           ;INITIAL TIMER VALUE
 501 030E                ADDCOR: .RES    1           ;ADDITION CORRECTION
 502 030F                CASFLG: .RES    1           ;CASSETTE MODE WHEN SET
 503 0310                TIMER2: .RES    2           ;FINAL TIMER VALUE. THESE TWO TIMER VALUES
 504                     ; ARE USED TO COMPUTE INTERVAL FOR BAUD RATE
 505 0312                TEMP1:  .RES    2           ;TEMPORARY STORAGE REGISTER
 506 0314                TEMP2:  .RES    1           ;TEMPORARY STORAGE REGISTER
 507 0315                TEMP3:  .RES    1           ;TEMPORARY STORAGE REGISTER
 508 0316                SAVIO:  .RES    1           ;SAVE SERIAL IN DATA PORT
 509 0317                TIMFLG: .RES    1           ;TIME OUT FLAG FOR BAUD RATE CORRECTION
 510 0318                STACKP: .RES    1           ;SIO STACK POINTER SAVE CELL
 511 0319                TSTAT:  .RES    1           ;TEMPORARY STATUS HOLDER
 512                     ;
 513                     ;
 514                     ;
 515 031A                HATABS: .RES    38          ;HANDLER ADDRESS TABLE
 516 0021                MAXDEV  =       *-HATABS-5  ;MAXIMUM HANDLER ADDRESS INDEX
 517                     ;
 518                     ;       NOTE : THE ENTIRE IOCB DEFINITIONS HAVE BEEN MODIFIED
 519                     ;
 520                     IOCB    *=      *           ;I/O CONTROL BLOCKS
 521 0340                ICHID:  .RES    1           ;HANDLER INDEX NUMBER (FF = IOCB FREE)
 522 0341                ICDNO:  .RES    1           ;DEVICE NUMBER (DRIVE NUMBER)
 523 0342                ICCOM:  .RES    1           ;COMMAND CODE
 524 0343                ICSTA:  .RES    1           ;STATUS OF LAST IOCB ACTION
 525 0344                ICBAL:  .RES    1           ;BUFFER ADDRESS LOW BYTE
 526 0345                ICBAH:  .RES    1
 527 0346                ICPTL:  .RES    1           ;PUT BYTE ROUTINE ADDRESS - 1
 528 0347                ICPTH:  .RES    1
 529 0348                ICBLL:  .RES    1           ;BUFFER LENGTH LOW BYTE
 530 0349                ICBLH:  .RES    1
 531 034A                ICAX1:  .RES    1           ;AUXILLARY INFORMATION FIRST BYTE
 532 034B                ICAX2:  .RES    1
 533 034C                ICSPR:  .RES    4           ;FOUR SPARE BYTES
 534 0350                        .RES    MAXIOC-IOCBSZ
 535                     ;
 536 03C0                PRNBUF: .RES    40          ;PRINTER BUFFER
 537                     ;
 538 03E8                        .RES    21          ;SPARE BYTES
 539                     ;
 540                     ;
 541                     ;
 542                     ;
 543                     ;
 544                     ;
 545                     ;
 546                     ;       PAGE FOUR RAM ASSIGNMENTS
 547                     ;
 548 03FD                CASBUF: .RES    131         ;CASSETTE BUFFER
```

```
549                             ;
550                             ; USER AREA STARTS HERE AND GOES TO END OF PAGE FIVE
551  0480                       USAREA: .RES    128             ;SPARE
552                             ;
553                             ;
554                             ;
555                             ;
556                             ;
557                             ;
558                             ;
559                             ;       PAGE FIVE RAM ASSIGNMENTS
560                             ;
561                             ;       PAGE FIVE IS RESERVED AS A USER WORK SPACE
562                             ;
563                             ;       NOTE: SEE FLOATING POINT SUBROUTINE AREA FOR PAGE FIVE CELLS
564                             ;
565                             ;
566                             ;       PAGE SIX RAM ASSIGNMENTS
567                             ;
568                             ;       PAGE SIX IS RESERVED AS A USER'S USER WORK SPACE
569                             ;
570                             ;
571                             ;
572                             ;
573                             ;       FLOATING POINT SUBROUTINES
574                             ;
575  0006                       FPREC   =       6               ;FLOATING PT PRECISION (# OF BYTES)
576                             ;
577                             ; IF CARRY USED THEN CARRY CLEAR => NO ERROR, CARR
578                             ;
579  D800                       AFP     =       $D800           ;ASCII->FLOATING POINT (FP)
580                             ;
581                             ; INBUFF+CIX -> FR0, CIX, CARRY
582                             ;
583  D8E6                       FASC    =       $D8E6           ;FP -> ASCII FR0-> LBUFF (INBUFF)
584  D9AA                       IFP     =       $D9AA           ;INTEGER -> FP
585                             ;
586                             ; 0-$FFFF (LSB,MSB) IN FR0,FR0+1->FR0
587                             ;
588  D9D2                       FPI     =       $D9D2           ;FP -> INTEGER FR0 -> FR0,FR0+1, CARRY
589  DA60                       FSUB    =       $DA60           ;FR0 <- FR0 - FR1 ,CARRY
590  DA66                       FADD    =       $DA66           ;FR0 <- FR0 + FR1 ,CARRY
591  DADB                       FMUL    =       $DADB           ;FR0 <- FR0 * FR1 ,CARRY
592  DB28                       FDIV    =       $DB28           ;FR0 <- FR0 / FR1 ,CARRY
593  DD89                       FLDOR   =       $DD89           ;FLOATING LOAD REG0 FR0 <- (X,Y)
594  DD8D                       FLDOP   =       $DD8D           ;    "        "   "   FR0 <- (FLPTR)
595  DD98                       FLD1R   =       $DD98           ;    "        "  REG1 FR1 <- (X,Y)
596  DD9C                       FLD1P   =       $DD9C           ;    "        "   "   FR1 <- (FLPTR)
597  DDA7                       FSTOR   =       $DDA7           ;FLOATING STORE REG0 (X,Y) <- FR0
598  DDAB                       FSTOP   =       $DDAB           ;    "        "     "   (FLPTR)<- FR0
599  DDB6                       FMOVE   =       $DDB6           ;FR1 <- FR0
600  DD40                       PLYEVL  =       $DD40           ;FR0 <- P(Z) = SUM(I=N TO 0) (A(I)*Z**I) CAR
601                             ;                                   INPUT:  (X,Y) = A(N),A(N-1)...A(0)  -> PL
602                             ;                                           ACC   = # OF COEFFICIENTS = DEGRE
```

```
                  ;                              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     =      $BEB1     ;FR0 <- SQUAREROOT(FR0) CARRY
                  ;
                  ;FLOATING POINT ROUTINES ZERO PAGE (NEEDED ONLY IF F.P. ROUTINES ARE CAL
                  ;
                          *=$D4
00D4              FR0:    .RES   FPREC     ;FP REG0
00DA              FRE:    .RES   FPREC
00E0              FR1:    .RES   FPREC     ;FP REG1
00E6              FR2:    .RES   FPREC
00EC              FRX:    .RES   1         ;FP SPARE
00ED              EEXP:   .RES   1         ;VALUE OF E
00EE              NSIGN:  .RES   1         ;SIGN OF #
00EF              ESIGN:  .RES   1         ;SIGN OF EXPONENT
00F0              FCHRFLG:.RES   1         ;1ST CHAR FLAG
00F1              DIGRT:  .RES   1         ;# OF DIGITS RIGHT OF DECIMAL
00F2              CIX:    .RES   1         ;CURRENT INPUT INDEX
00F3              INBUFF: .RES   2         ;POINTS TO USER'S LINE INPUT BUFFER
00F5              ZTEMP1: .RES   2
00F7              ZTEMP4: .RES   2
00F9              ZTEMP3: .RES   2
00FB              DEGFLG
00FB              RADFLG: .RES   1         ;0=RADIANS, 6=DEGREES
0000              RADON   =      0         ;INDICATES RADIANS
0006              DEGON   =      6         ;INDICATES DEGREES
00FC              FLPTR:  .RES   2         ;POINTS TO USER'S FLOATING PT NUMBER
00FE              FPTR2:  .RES   2
                  ;
                  ;FLOATING PT ROUTINES' NON-ZERO PAGE RAM (NEEDED ONLY IF F.P. ROUTINES C
                  ;
                          *=$57E
057E              LBPR1:  .RES   1         ;LBUFF PREFIX 1
057F              LBPR2:  .RES   1         ;LBUFF PREFIX 2
0580              LBUFF:  .RES   128       ;LINE BUFFER
05E0              PLYARG  =      LBUFF+$60 ;POLYNOMIAL ARGUMENTS
05E6              FPSCR   =      PLYARG+FPREC
05EC              FPSCR1  =      FPSCR+FPREC
05E6              FSCR    =      FPSCR
05EC              FSCR1   =      FPSCR1
05FF              LBFEND  =      *-1       ;END OF LBUFF
                  ;
                  ;
                  ;
                  ;
```

```
                                ;
                                ;
                                ;
                                ;
                                ;       COLLEEN MNEMONICS
                                ;
D200                            POKEY   =       $D200       ;VBLANK ACTION:     DESCRIPTION:
D200                            POT0    =       POKEY+0     ;POT0-->PADDL0      0-227 IN RAM CELL
D201                            POT1    =       POKEY+1     ;POT1-->PADDL1      0-227 IN RAM CELL
D202                            POT2    =       POKEY+2     ;POT2-->PADDL2      0-227 IN RAM CELL
D203                            POT3    =       POKEY+3     ;POT3-->PADDL3      0-227 IN RAM CELL
D204                            POT4    =       POKEY+4     ;POT4-->PADDL4      0-227 IN RAM CELL
D205                            POT5    =       POKEY+5     ;POT5-->PADDL5      0-227 IN RAM CELL
D206                            POT6    =       POKEY+6     ;POT6-->PADDL6      0-227 IN RAM CELL
D207                            POT7    =       POKEY+7     ;POT7-->PADDL7      0-227 IN RAM CELL
D208                            ALLPOT  =       POKEY+8     ;???
D209                            KBCODE  =       POKEY+9
D20A                            RANDOM  =       POKEY+10
D20B                            POTGO   =       POKEY+11    ;STROBED
D20D                            SERIN   =       POKEY+13
D20E                            IRQST   =       POKEY+14
D20F                            SKSTAT  =       POKEY+15
D200                            AUDF1   =       POKEY+0
D201                            AUDC1   =       POKEY+1
D202                            AUDF2   =       POKEY+2
D203                            AUDC2   =       POKEY+3
D204                            AUDF3   =       POKEY+4
D205                            AUDC3   =       POKEY+5
D206                            AUDF4   =       POKEY+6
D207                            AUDC4   =       POKEY+7
D208                            AUDCTL  =       POKEY+8     ;NONE               AUDCTL<--[SIO]
D209                            STIMER  =       POKEY+9
D20A                            SKRES   =       POKEY+10    ;NONE               SKRES<--[SIO]
D20D                            SEROUT  =       POKEY+13    ;NONE               SEROUT<--[SIO]
D20E                            IRGEN   =       POKEY+14    ;POKMSK-->IRQEN     (AFFECTED BY OPEN S: OR
D20F                            SKCTL   =       POKEY+15    ;SSKCTL-->SKCTL     SSKCTL<--[SIO]
                                ;
D000                            CTIA    =       $D000       ;VBLANK ACTION:     DESCRIPTION:
D000                            HPOSP0  =       CTIA+0
D001                            HPOSP1  =       CTIA+1
D002                            HPOSP2  =       CTIA+2
D003                            HPOSP3  =       CTIA+3
D004                            HPOSM0  =       CTIA+4
D005                            HPOSM1  =       CTIA+5
D006                            HPOSM2  =       CTIA+6
D007                            HPOSM3  =       CTIA+7
D008                            SIZEP0  =       CTIA+8
D009                            SIZEP1  =       CTIA+9
D00A                            SIZEP2  =       CTIA+10
D00B                            SIZEP3  =       CTIA+11
D00C                            SIZEM   =       CTIA+12
D00D                            GRAFP0  =       CTIA+13
D00E                            GRAFP1  =       CTIA+14
```

```
    711 D00F                    GRAFP2  =       CTIA+15
    712 D010                    GRAFP3  =       CTIA+16
    713 D011                    GRAFM   =       CTIA+17
    714 D012                    COLPM0  =       CTIA+18     ;PCOLR0-->COLPM0    WITH ATTRACT MODE
    715 D013                    COLPM1  =       CTIA+19     ;PCOLR1-->COLPM1    WITH ATTRACT MODE
    716 D014                    COLPM2  =       CTIA+20     ;PCOLR2-->COLPM2    WITH ATTRACT MODE
    717 D015                    COLPM3  =       CTIA+21     ;PCOLR3-->COLPM3    WITH ATTRACT MODE
    718 D016                    COLPF0  =       CTIA+22     ;COLOR0-->COLPF0    WITH ATTRACT MODE
    719 D017                    COLPF1  =       CTIA+23     ;COLOR1-->COLPF1    WITH ATTRACT MODE
    720 D018                    COLPF2  =       CTIA+24     ;COLOR2-->COLPF2    WITH ATTRACT MODE
    721 D019                    COLPF3  =       CTIA+25     ;COLOR3-->COLPF3    WITH ATTRACT MODE
    722 D01A                    COLBK   =       CTIA+26     ;COLOR4-->COLBK     WITH ATTRACT MODE
    723 D01B                    PRIOR   =       CTIA+27     ;(ON OPEN S: OR E:) GPRIOR-->PRIOR
    724 D01C                    VDELAY  =       CTIA+28
    725 D01D                    GRACTL  =       CTIA+29
    726 D01E                    HITCLR  =       CTIA+30
    727 D01F                    CONSOL  =       CTIA+31     ;$08-->CONSOL       TURN OFF SPEAKER
    728 D000                    M0PF    =       CTIA+0
    729 D001                    M1PF    =       CTIA+1
    730 D002                    M2PF    =       CTIA+2
    731 D003                    M3PF    =       CTIA+3
    732 D004                    P0PF    =       CTIA+4
    733 D005                    P1PF    =       CTIA+5
    734 D006                    P2PF    =       CTIA+6
    735 D007                    P3PF    =       CTIA+7
    736 D008                    M0PL    =       CTIA+8
    737 D009                    M1PL    =       CTIA+9
    738 D00A                    M2PL    =       CTIA+10
    739 D00B                    M3PL    =       CTIA+11
    740 D00C                    P0PL    =       CTIA+12
    741 D00D                    P1PL    =       CTIA+13
    742 D00E                    P2PL    =       CTIA+14
    743 D00F                    P3PL    =       CTIA+15
    744 D010                    TRIG0   =       CTIA+16     ;TRIG0-->STRIG0
    745 D011                    TRIG1   =       CTIA+17     ;TRIG1-->STRIG1
    746 D012                    TRIG2   =       CTIA+18     ;TRIG2-->STRIG2
    747 D013                    TRIG3   =       CTIA+19     ;TRIG3-->STRIG3
    748                         ;
    749 D400                    ANTIC   =       $D400       ;VBLANK ACTION      DESCRIPTION
    750 D400                    DMACTL  =       ANTIC+0     ;DMACTL<--SDMCTL    ON OPEN S: OR E:
    751 D401                    CHACTL  =       ANTIC+1     ;CHACTL<--CHACT     ON OPEN S: OR E:
    752 D402                    DLISTL  =       ANTIC+2     ;DLISTL<--SDLSTL    ON OPEN S: OR E:
    753 D403                    DLISTH  =       ANTIC+3     ;DLISTH<--SDLSTH    ON OPEN S: OR E:
    754 D404                    HSCROL  =       ANTIC+4
    755 D405                    VSCROL  =       ANTIC+5
    756 D407                    PMBASE  =       ANTIC+7
    757 D409                    CHBASE  =       ANTIC+9     ;CHBASE<--CHBAS     ON OPEN S: OR E:
    758 D40A                    WSYNC   =       ANTIC+10
    759 D40B                    VCOUNT  =       ANTIC+11
    760 D40C                    PENH    =       ANTIC+12
    761 D40D                    PENV    =       ANTIC+13
    762 D40E                    NMIEN   =       ANTIC+14    ;NMIEN<--40         POWER ON AND [SETVBV]
    763 D40F                    NMIRES  =       ANTIC+15    ;STROBED
    764 D40F                    NMIST   =       ANTIC+15
```

```
      765  D300                PIA     =       $D300       ;VBLANK ACTION      DESCRIPTION
      766  D300                PORTA   =       PIA+0       ;PORTA-->STICK0,1   X-Y CONTROLLERS
      767  D301                PORTB   =       PIA+1       ;PORTB-->STICK2,3   X-Y CONTROLLERS
      768  D302                PACTL   =       PIA+2       ;NONE               PACTL<--3C [INIT]
      769  D303                PBCTL   =       PIA+3       ;NONE               PBCTL<--3C [INIT]
      770                      ;
      771                      ;
      772                      ;
      773                      ; .PAGE
```

```
774                             .PAGE
775                             LIST    S
776                              TITLE  'CENTRAL INPUT/OUTPUT (CIO) 2-7-79'
777                     ;               UPDATED BY AL MILLER 3-9-79
778  0030               ASCZER  =       '0              ;ASCII ZERO
779  003A               COLON   =       $3A             ;ASCII COLON
780  009B               EOL     =       $9B             ;END OF RECORD
```

```
 781                            .PAGE
 782                    ;
 783                    ; CIO JUMP VECTOR FOR USERS
 784                            *=CIOV
 785  E456  4C C4 E4           JMP     CIO             ;GO TO CIO
 786                    ;
 787                    ; CIO INIT JUMP VECTOR FOR POWER UP
 788                            *=CIOINV
 789  E46E  4C A6 E4           JMP     CIOINT          ;GO TO INIT
 790                    ;
 791                    ;
 792                    ; ERROR ROUTINE ADDRESS EQUATE
 793                    ;ERRTNH=ERRTN/256              "MOVED TO LINE 788"
 794                    ;ERRTNL=-ERRTNH*256+ERRTN      "MOVED TO LINE 789"
 795                    ;
 796                    ;
 797                            *=CIOORG
 798                    ;
 799                    ; CIO INITIALIZATION (CALLED BY MONITOR AT POWER UP)
 800  E4A6  A2 00       CIOINT: LDX     #0
 801  E4A8  A9 FF       CIOI1:  LDA     #IOCFRE         ;SET ALL IOCB'S TO FREE
 802  E4AA  9D 40 03            STA     ICHID,X         ;BY SETTING HANDLER ID'S=$FF
 803  E4AD  A9 C0               LDA     #ERRTNL
 804  E4AF  9D 46 03            STA     ICPTL,X         ;POINT PUT TO ERROR ROUTINE
 805  E4B2  A9 E4               LDA     #ERRTNH
 806  E4B4  9D 47 03            STA     ICPTH,X
 807  E4B7  8A                  TXA
 808  E4B8  18                  CLC
 809  E4B9  69 10               ADC     #IOCBSZ         ;BUMP INDEX BY SIZE
 810  E4BB  AA                  TAX
 811  E4BC  C9 80               CMP     #MAXIOC         ;DONE?
 812  E4BE  90 E8               BCC     CIOI1           ;NO
 813  E4C0  60                  RTS                     ;YES, RETURN
 814                    ;
 815                    ; ERROR ROUTINE FOR ILLEGAL PUT
 816  E4C0              ERRTN=*-1
 817  00E4              ERRTNH=ERRTN/256
 818  00C0              ERRTNL=(-ERRTNH)*256+ERRTN
 819  E4C1  A0 85               LDY     #NOTOPN         ;IOCB NOT OPEN
 820  E4C3  60                  RTS
```

```
 821                            .PAGE
 822                    ;
 823                    ; CIO LOCAL RAM (USES SPARE BYTES IN ZERO PAGE IOCB)
 824  002C              ENTVEC  =       ICSPRZ
 825                    ;
 826                    ; CIO MAIN ROUTINE
 827                    ;
 828                    ; CIO INTERFACES BETWEEN USER AND INPUT/OUTPUT DE
 829  E4C4  85 2F       CIO:    STA     CIOCHR          ;SAVE POSSIBLE OUTPUT CHARACTER
 830  E4C6  86 2E             STX     ICIDNO          ;SAVE IOCB NUMBER * N
 831                    ;
 832                    ; CHECK FOR LEGAL IOCB
 833  E4C8  8A                TXA
 834  E4C9  29 0F             AND     #$F             ;IS IOCB MULTIPLE OF 16?
 835  E4CB  D0 04             BNE     CIERR1          ;NO, ERROR
 836  E4CD  E0 80             CPX     #MAXIOC         ;IS INDEX TOO LARGE?
 837  E4CF  90 05             BCC     IOC1            ;NO
 838                    ;
 839                    ; INVALID IOCB NUMBER -- RETURN ERROR
 840  E4D1  A0 86       CIERR1: LDY     #BADIOC         ;ERROR CODE
 841  E4D3  4C 1B E6          JMP     CIRTN1          ;RETURN
 842                    ;
 843                    ; MOVE USER IOCB TO ZERO PAGE
 844  E4D6  A0 00       IOC1:   LDY     #0
 845  E4D8  BD 40 03    IOC1A:  LDA     IOCB,X          ;USER IOCB
 846  E4DB  99 20 00          STA     IOCBAS,Y        ;TO ZERO PAGE
 847  E4DE  E8                INX                     ;
 848  E4DF  C8                INY                     ;
 849  E4E0  C0 0C             CPY     #12             ;12 BYTES
 850  E4E2  90 F4             BCC     IOC1A
 851                    ;
 852                    ; COMPUTE CIO INTERNAL VECTOR FOR COMMAND
 853  E4E4  A0 84             LDY     #NVALID         ;ASSUME INVALID CODE
 854  E4E6  A5 22             LDA     ICCOMZ          ;COMMAND CODE TO INDEX
 855  E4E8  C9 03             CMP     #OPEN           ;IS COMMAND LEGAL?
 856  E4EA  90 25             BCC     CIERR4          ;NO
 857  E4EC  A8                TAY
 858                    ;
 859                    ; MOVE COMMAND TO ZERO BASE FOR INDEX
 860  E4ED  C0 0E             CPY     #SPECIL         ;IS COMMAND SPECIAL?
 861  E4EF  90 02             BCC     IOC2            ;NO
 862  E4F1  A0 0E             LDY     #SPECIL         ;YES, SET SPECIAL OFFSET INDEX
 863  E4F3  84 17       IOC2:   STY     ICCOMT          ;SAVE COMMAND FOR VECTOR
 864  E4F5  B9 C6 E6          LDA     COMTAB-3,Y      ;GET VECTOR OFFSET FROM TABLE
 865  E4F8  F0 0F             BEQ     CIOPEN          ;GO IF OPEN COMMAND
 866  E4FA  C9 02             CMP     #2              ;IS IT CLOSE?
 867  E4FC  F0 05             BEQ     CICLOS          ;YES
 868  E4FE  C9 08             CMP     #8              ;IS IT STATUS OR SPECIAL?
 869  E500  B0 4C             BCS     CISTSP          ;YES
 870  E502  C9 04             CMP     #4              ;IS IT READ?
 871  E504  F0 63             BEQ     CIREAD          ;YES
 872  E506  4C C9 E5          JMP     CIWRIT          ;ELSE, MUST BE WRITE
```

```
 873                            .PAGE
 874                    ;
 875                    ; OPEN COMMAND
 876                    ;
 877                    ; FIND DEVICE HANDLER IN HANDLER ADDRESS TABLE
 878  E509  A5 20       CIOPEN: LDA     ICHIDZ          ;GET HANDLER ID
 879  E50B  C9 FF               CMP     #IOCFRE         ;IS THIS IOCB CLOSED?
 880  E50D  F0 05               BEQ     IOC6            ;YES
 881                    ;
 882                    ; ERROR -- IOCB ALREADY OPEN
 883  E50F  A0 81       CIERR3: LDY     #PRVOPN         ;ERROR CODE
 884  E511  4C 1B E6    CIERR4: JMP     CIRTN1          ;RETURN
 885                    ;
 886                    ; GO FIND DEVICE
 887  E514  20 9E E6    IOC6:   JSR     DEVSRC          ;CALL DEVICE SEARCH
 888  E517  B0 F8               BCS     CIERR4          ;GO IF DEVICE NOT FOUND
 889                    ;
 890                    ; DEVICE FOUND, INITIALIZE IOCB FOR OPEN
 891                    ;
 892                    ; COMPUTE HANDLER ENTRY POINT
 893  E519  20 3D E6    IOC7:   JSR     COMENT
 894  E51C  B0 F3               BCS     CIERR4          ;GO IF ERROR IN COMPUTE
 895                    ;
 896                    ; GO TO HANDLER FOR INITIALIZATION
 897  E51E  20 89 E6            JSR     GOHAND          USE INDIRECT JUMP
 898                    ;
 899                    ; STORE PUT BYTE ADDRESS-1 INTO IOCB
 900  E521  A9 0B               LDA     #PUTCHR         ;SIMULATE PUT CHARACTER
 901  E523  85 17               STA     ICCOMT
 902  E525  20 3D E6            JSR     COMENT          ;COMPUTE ENTRY POINT
 903  E528  A5 2C               LDA     ICSPRZ          ;MOVE COMPUTED VALUE
 904  E52A  85 26               STA     ICPTLZ          ;TO PUT BYTE ADDRESS
 905  E52C  A5 2D               LDA     ICSPRZ+1
 906  E52E  85 27               STA     ICPTHZ
 907  E530  4C 1D E6            JMP     CIRTN2          ;RETURN TO USER
```

```
908                             .PAGE
909                     ;
910                     ;
911                     ; CLOSE COMMAND
912  E533  A0 01        CICLOS: LDY     #SUCCES     ;ASSUME GOOD CLOSE
913  E535  84 23                STY     ICSTAZ
914  E537  20 3D E6             JSR     COMENT      ;COMPUTE HANDLER ENTRY POINT
915  E53A  B0 03                BCS     CICLO2      ;GO IF ERROR IN COMPUTE
916  E53C  20 89 E6             JSR     GOHAND      ;GO TO HANDLER TO CLOSE DEVICE
917  E53F  A9 FF        CICLO2: LDA     #IOCFRE     ;GET IOCB "FREE" VALUE
918  E541  85 20                STA     ICHIDZ      ;SET HANDLER ID
919  E543  A9 E4                LDA     #ERRTNH
920  E545  85 27                STA     ICPTHZ      ;SET PUT BYTE TO POINT TO ERROR
921  E547  A9 C0                LDA     #ERRTNL
922  E549  85 26                STA     ICPTLZ
923  E54B  4C 1D E6             JMP     CIRTN2      ;RETURN
924                     ;
925                     ;
926                     ; STATUS AND SPECIAL REQUESTS
927                     ; DO IMPLIED OPEN IF NECESSARY AND GO TO DEVICE
928  E54E  A5 20        CISTSP: LDA     ICHIDZ      ;IS THERE A HANDLER ID?
929  E550  C9 FF                CMP     #IOCFRE
930  E552  D0 05                BNE     CIST1       ;YES
931                     ;
932                     ; IOCB IS FREE, DO IMPLIED OPEN
933  E554  20 9E E6             JSR     DEVSRC      ;FIND DEVICE IN TABLE
934  E557  B0 B8                BCS     CIERR4      ;GO IF ERROR IN COMPUTE
935                     ;
936                     ; COMPUTE AND GO TO ENTRY POINT IN HANDLER
937  E559  20 3D E6     CIST1:  JSR     COMENT      ;COMPUTER HANDLER ENTRY VECTOR
938  E55C  20 89 E6             JSR     GOHAND      ;GO TO HANDLER
939                     ;
940                     ; RESTORE HANDLER INDEX (DO IMPLIED CLOSE)
941  E55F  A6 2E                LDX     ICIDNO      ;IOCB INDEX
942  E561  BD 40 03             LDA     ICHID,X     ;GET ORIGINAL HANDLER ID
943  E564  85 20                STA     ICHIDZ      ;RESTORE ZERO PAGE
944  E566  4C 1D E6             JMP     CIRTN2      ;RETURN
```

```
945                             .PAGE
946                     ;
947                     ; READ -- DO GET COMMANDS
948  E569  A5 22        CIREAD: LDA     ICCOMZ          ;GET COMMAND BYTE
949  E56B  25 2A                AND     ICAX1Z          ;IS THIS READ LEGAL?
950  E56D  D0 05                BNE     RCI1A           ;YES
951                     ;
952                     ; ILLEGAL READ -- IOCB OPENED FOR WRITE ONLY
953  E56F  A0 83                LDY     #WRONLY         ;ERROR CODE
954  E571  4C 1B E6     RCI1B:  JMP     CIRTN1          ;RETURN
955                     ;
956                     ; COMPUTE AND CHECK ENTRY POINT
957  E574  20 3D E6     RCI1A:  JSR     COMENT          ;COMPUTE ENTRY POINT
958  E577  B0 F8                BCS     RCI1B           ;GO IF ERROR IN COMPUTE
959                     ;
960                     ; GET RECORD OR CHARACTERS
961  E579  A5 28                LDA     ICBLLZ
962  E57B  05 29                ORA     ICBLLZ+1        ;IS BUFFER LENGTH ZERO?
963  E57D  D0 08                BNE     RCI3            ;NO
964  E57F  20 89 E6             JSR     GOHAND
965  E582  85 2F                STA     CIOCHR
966  E584  4C 1D E6             JMP     CIRTN2
967                     ;
968                     ; LOOP TO FILL BUFFER OR END RECORD
969  E587  20 89 E6     RCI3:   JSR     GOHAND          ;GO TO HANDLER TO GET BYTE
970  E58A  85 2F                STA     CIOCHR          ;SAVE BYTE
971  E58C  30 35                BMI     RCI4            ;END TRANSFER IF ERROR
972  E58E  A0 00                LDY     #0
973  E590  91 24                STA     (ICBALZ),Y      ;PUT BYTE IN USER BUFFER
974  E592  20 70 E6             JSR     INCBFP          ;INCREMENT BUFFER POINTER
975  E595  A5 22                LDA     ICCOMZ          ;GET COMMAND CODE
976  E597  29 02                AND     #2              ;IS IT GET RECORD?
977  E599  D0 0C                BNE     RCI1            ;NO
978                     ;
979                     ; CHECK FOR EOL ON TEXT RECORDS
980  E59B  A5 2F                LDA     CIOCHR          ;GET BYTE
981  E59D  C9 9B                CMP     #EOL            ;IS IT AN EOL?
982  E59F  D0 06                BNE     RCI1            ;NO
983  E5A1  20 63 E6             JSR     DECBFL          ;YES, DECREMENT BUFFER LENGTH
984  E5A4  4C C3 E5             JMP     RCI4            ;END TRANSFER
985                     ;
986                     ; CHECK BUFFER FULL
987  E5A7  20 63 E6     RCI1:   JSR     DECBFL          ;DECREMENT BUFFER LENGTH
988  E5AA  D0 DB                BNE     RCI3            ;CONTINUE IF NON ZERO
```

```
 989                            .PAGE
 990                    ;
 991                    ; BUFFER FULL, RECORD NOT ENDED
 992                    ; DISCARD BYTES UNTIL END OF RECORD
 993 E5AC A5 22         RCI2:   LDA     ICCOMZ          ;GET COMMAND BYTE
 994 E5AE 29 02                 AND     #2              ;IS IT GET CHARACTER?
 995 E5B0 D0 11                 BNE     RCI4            ;YES, END TRANSFER
 996                    ;
 997                    ; LOOP TO WAIT FOR EOL
 998 E5B2 20 89 E6      RCI6:   JSR     GOHAND          ;GET BYTE FROM HANDLER
 999 E5B5 85 2F                 STA     CIOCHR          ;SAVE CHARACTER
1000 E5B7 30 0A                 BMI     RCI4            ;GO IF ERROR
1001                    ;
1002                    ; TEXT RECORD, WAIT FOR EOL
1003 E5B9 A5 2F                 LDA     CIOCHR          ;GET GOT BYTE
1004 E5BB C9 9B                 CMP     #EOL            ;IS IT EOL?
1005 E5BD D0 F3                 BNE     RCI6            ;NO, CONTINUE
1006                    ;
1007                    ; END OF RECORD, BUFFER FULL -- SEND TRUNCATED RECORD MESSAGE
1008 E5BF A9 89         RCI11:  LDA     #TRNRCD         ;ERROR CODE
1009 E5C1 85 23                 STA     ICSTAZ          ;STORE IN IOCB
1010                    ;
1011                    ; TRANSFER DONE
1012 E5C3 20 77 E6      RCI4:   JSR     SUBBFL          ;SET FINAL BUFFER LENGTH
1013 E5C6 4C 1D E6              JMP     CIRTN2          ;RETURN
```

```
1014                            .PAGE
1015                    ;
1016                    ; WRITE -- DO PUT COMMANDS
1017  E5C9  A5 22       CIWRIT: LDA     ICCOMZ          ;GET COMMAND BYTE
1018  E5CB  25 2A               AND     ICAX1Z          ;IS THIS WRITE LEGAL?
1019  E5CD  D0 05               BNE     WCI1A           ;YES
1020                    ;
1021                    ; ILLEGAL WRITE -- DEVICE OPENED FOR READ ONLY
1022  E5CF  A0 87               LDY     #RDONLY         ;ERROR CODE
1023  E5D1  4C 1B E6    WCI1B:  JMP     CIRTN1          ;RETURN
1024                    ;
1025                    ; COMPUTE AND CHECK ENTRY POINT
1026  E5D4  20 3D E6    WCI1A:  JSR     COMENT          ;COMPUTE HANDLER ENTRY POINT
1027  E5D7  B0 F8               BCS     WCI1B           ;GO IF ERROR IN COMPUTE
1028                    ;
1029                    ; PUT RECORD OR CHARACTERS
1030  E5D9  A5 28               LDA     ICBLLZ
1031  E5DB  05 29               ORA     ICBLLZ+1        ;IS BUFFER LENGTH ZERO?
1032  E5DD  D0 06               BNE     WCI3            ;NO
1033  E5DF  A5 2F               LDA     CIOCHR          ;GET CHARACTER
1034  E5E1  E6 28               INC     ICBLLZ          ;SET BUFFER LENGTH=1
1035  E5E3  D0 06               BNE     WCI4            ;THEN JUST TRANSFER ONE BYTE
1036                    ;
1037                    ; LOOP TO TRANSFER BYTES FROM BUFFER TO HANDLER
1038  E5E5  A0 00       WCI3:   LDY     #0
1039  E5E7  B1 24               LDA     (ICBALZ),Y      ;GET BYTE FROM BUFFER
1040  E5E9  85 2F               STA     CIOCHR          ;SAVE
1041  E5EB  20 89 E6    WCI4:   JSR     GOHAND          ;GO PUT BYTE
1042  E5EE  30 25               BMI     WCI5            ;END IF ERROR
1043  E5F0  20 70 E6            JSR     INCBFP          ;INCREMENT BUFFER POINTER
1044                    ;
1045                    ; CHECK FOR TEXT RECORD
1046  E5F3  A5 22               LDA     ICCOMZ          ;GET COMMAND BYTE
1047  E5F5  29 02               AND     #2              ;IS IT PUT RECORD?
1048  E5F7  D0 0C               BNE     WCI1            ;NO
1049                    ;
1050                    ; TEXT RECORD -- CHECK FOR EOL TRANSFER
1051  E5F9  A5 2F               LDA     CIOCHR          ;GET LAST CHARACTER
1052  E5FB  C9 9B               CMP     #EOL            ;IS IT AN EOL?
1053  E5FD  D0 06               BNE     WCI1            ;NO
1054  E5FF  20 63 E6            JSR     DECBFL          ;DECREMENT BUFFER LENGTH
1055  E602  4C 15 E6            JMP     WCI5            ;END TRANSFER
1056                    ;
1057                    ; CHECK FOR BUFFER EMPTY
1058  E605  20 63 E6    WCI1:   JSR     DECBFL          ;DECREMENT BUFFER LENGTH
1059  E608  D0 DB               BNE     WCI3            ;CONTINUE IF NON ZERO
```

```
1060                            .PAGE
1061                    ;
1062                    ; BUFFER EMPTY, RECORD NOT FILLED
1063                    ; CHECK TYPE OF TRANSFER
1064  E60A  A5 22       WCI2:   LDA     ICCOMZ          ;GET COMMAND CODE
1065  E60C  29 02               AND     #2              ;IS IT PUT CHARACTER?
1066  E60E  D0 05               BNE     WCI5            ;YES, END TRANSFER
1067                    ;
1068                    ; PUT RECORD (TEXT), BUFFER EMPTY, SEND EOL
1069  E610  A9 9B               LDA     #EOL
1070  E612  20 89 E6            JSR     GOHAND          ;GO TO HANDLER
1071                    ;
1072                    ; END PUT TRANSFER
1073  E615  20 77 E6    WCI5:   JSR     SUBBFL          ;SET ACTUAL PUT BUFFER LENGTH
1074  E618  4C 1D E6            JMP     CIRTN2          ;RETURN
```

```
1075                            .PAGE
1076                    ;
1077                    ; CIO RETURNS
1078                    ; RETURNS WITH Y=STATUS
1079  E61B  84 23       CIRTN1: STY     ICSTAZ          ;SAVE STATUS
1080                    ;
1081                    ; RETURNS WITH STATUS STORED IN ICSTAZ
1082                    ; MOVE IOCB IN ZERO PAGE BACK TO USER AREA
1083                    ;
1084  E61D  A4 2E       CIRTN2: LDY     ICIDNO          ;GET IOCB INDEX
1085  E61F  B9 44 03            LDA     ICBAL,Y
1086  E622  85 24               STA     ICBALZ          ;RESTORE USER BUFFER POINTER
1087  E624  B9 45 03            LDA     ICBAH,Y
1088  E627  85 25               STA     ICBAHZ
1089  E629  A2 00               LDX     #0              ;LOOP COUNT AND INDEX
1090  E62B  B5 20       CIRT3:  LDA     IOCBAS,X        ;ZERO PAGE
1091  E62D  99 40 03            STA     IOCB,Y          ;TO USER AREA
1092  E630  E8                 INX
1093  E631  C8                 INY
1094  E632  E0 0C               CPX     #12             ;12 BYTES
1095  E634  90 F5               BCC     CIRT3
1096                    ;
1097                    ; RESTORE A,X, & Y
1098                    ;
1099  E636  A5 2F               LDA     CIOCHR          ;GET LAST CHARACTER
1100  E638  A6 2E               LDX     ICIDNO          ;IOCB INDEX
1101  E63A  A4 23               LDY     ICSTAZ          ;GET STATUS AND SET FLAGS
1102  E63C  60                  RTS                     ;RETURN TO USER
```

```
1103                            .PAGE
1104                    ;
1105                    ;
1106                    ; CIO SUBROUTINES
1107                    ;
1108                    ; COMENT -- CHECK AND COMPUTE HANDLER ENTRY POINT
1109  E63D  A4 20       COMENT: LDY     ICHIDZ          ;GET HANDLER INDEX
1110  E63F  C0 22               CPY     #MAXDEV+1       ;IS IT A LEGAL INDEX?
1111  E641  90 04               BCC     COM1            ;YES
1112                    ;
1113                    ; ILLEGAL HANDLER INDEX MEANS DEVICE NOT OPEN FOR OPERATION
1114  E643  A0 85               LDY     #NOTOPN         ;ERROR CODE
1115  E645  B0 1B               BCS     COM2            ;RETURN
1116                    ;
1117                    ; USE HANDLER ADDRESS TABLE AND COMMAND TABLE TO GET VECTOR
1118  E647  B9 1B 03    COM1:   LDA     HATABS+1,Y      ;GET LOW BYTE OF ADDRESS
1119  E64A  85 2C               STA     ICSPRZ          ;AND SAVE IN POINTER
1120  E64C  B9 1C 03            LDA     HATABS+2,Y      ;GET HI BYTE OF ADDRESS
1121  E64F  85 2D               STA     ICSPRZ+1
1122  E651  A4 17               LDY     ICCOMT          ;GET COMMAND CODE
1123  E653  B9 C6 E6            LDA     COMTAB-3,Y      ;GET COMMAND OFFSET
1124  E656  A8                  TAY
1125  E657  B1 2C               LDA     (ICSPRZ),Y      ;GET LOW BYTE OF VECTOR FROM
1126  E659  AA                  TAX                     ;HANDLER ITSELF AND SAVE
1127  E65A  C8                  INY
1128  E65B  B1 2C               LDA     (ICSPRZ),Y      ;GET HI BYTE OF VECTOR
1129  E65D  85 2D               STA     ICSPRZ+1
1130  E65F  86 2C               STX     ICSPRZ          ;SET LO BYTE
1131  E661  18                  CLC                     ;SHOW NO ERROR
1132  E662  60          COM2:   RTS
1133                    ;
1134                    ;
1135                    ; DECBFL -- DECREMENT BUFFER LENGTH DOUBLE BYTE
1136                    ; Z FLAG = 0 ON RETURN IF LENGTH = 0 AFTER DECREMENT
1137  E663  C6 28       DECBFL: DEC     ICBLLZ          ;DECREMENT LOW BYTE
1138  E665  A5 28               LDA     ICBLLZ          ;CHECK IT
1139  E667  C9 FF               CMP     #$FF            ;DID IT GO BELOW?
1140  E669  D0 02               BNE     DECBF1          ;NO
1141  E66B  C6 29               DEC     ICBLLZ+1        ;DECREMENT HI BYTE
1142  E66D  05 29       DECBF1: ORA     ICBLLZ+1        ;SET Z IF BOTH ARE ZERO
1143  E66F  60                  RTS
1144                    ;
1145                    ;
1146                    ; INCBFP -- INCREMENT WORKING BUFFER POINTER
1147  E670  E6 24       INCBFP: INC     ICBALZ          ;BUMP LOW BYTE
1148  E672  D0 02               BNE     INCBF1          ;GO IF NOT ZERO
1149  E674  E6 25               INC     ICBALZ+1        ;ELSE, BUMP HI BYTE
1150  E676  60          INCBF1: RTS
1151                    ;
1152                    ;
1153                    ; SUBBFL -- SET BUFFER LENGTH = BUFFER LENGTH - WORKING BYTE COUNT
1154  E677  A6 2E       SUBBFL: LDX     ICIDNO          ;GET IOCB INDEX
1155  E679  38                 SEC
1156  E67A  BD 48 03            LDA     ICBLL,X         ;GET LOW BYTE OF INITIAL LENGTH
```

```
1157  E67D  E5 28              SBC     ICBLLZ      ;SUBTRACT FINAL LOW BYTE
1158  E67F  85 28              STA     ICBLLZ      ;AND SAVE BACK
1159  E681  BD 49 03           LDA     ICBLH,X     ;GET HI BYTE
1160  E684  E5 29              SBC     ICBLLZ+1
1161  E686  85 29              STA     ICBLHZ
1162  E688  60                 RTS
1163                        ;
1164                        ;
1165                        ; GOHAND -- GO INDIRECT TO A DEVICE HANDLER
1166                        ; Y= STATUS ON RETURN, N FLAG=1 IF ERROR ON RETURN
1167  E689  A0 92           GOHAND: LDY     #FNCNOT     ;PREPARE NO FUNCTION STATUS FOR HANDLER RTS
1168  E68B  20 93 E6              JSR     CIJUMP      ;USE THE INDIRECT JUMP
1169  E68E  84 23                 STY     ICSTAZ      ;SAVE STATUS
1170  E690  C0 00                 CPY     #0          ;AND SET N FLAG
1171  E692  60                    RTS
1172                        ;
1173                        ; INDIRECT JUMP TO HANDLER BY PAUL'S METHOD
1174  E693  AA              CIJUMP: TAX                 ;SAVE A
1175  E694  A5 2D                 LDA     ICSPRZ+1    ;GET JUMP ADDRESS HI BYTE
1176  E696  48                    PHA                 ;PUT ON STACK
1177  E697  A5 2C                 LDA     ICSPRZ      ;GET JUMP ADDRESS LO BYTE
1178  E699  48                    PHA                 ;PUT ON STACK
1179  E69A  8A                    TXA                 ;RESTORE A
1180  E69B  A6 2E                 LDX     ICIDNO      ;GET IOCB INDEX
1181  E69D  60                    RTS                 ;GO TO HANDLER INDIRECTLY
```

```
1182                            .PAGE
1183                    ;
1184                    ; DEVSRC -- DEVICE SEARCH, FIND DEVICE IN HANDLER ADDRESS TABLE
1185                    ;
1186                    ; LOOP TO FIND DEVICE
1187  E69E  A0 00       DEVSRC: LDY     #0
1188  E6A0  B1 24               LDA     (ICBALZ),Y  ;GET DEVICE NAME FROM USER
1189  E6A2  F0 0C               BEQ     CIERR2
1190  E6A4  A0 21               LDY     #MAXDEV     ;INITIAL COMPARE INDEX
1191  E6A6  D9 1A 03    DEVS1:  CMP     HATABS,Y    ;IS THIS THE DEVICE?
1192  E6A9  F0 0A               BEQ     DEVS2       ;YES
1193  E6AB  88                  DEY
1194  E6AC  88                  DEY                 ;ELSE, POINT TO NEXT DEVICE NAME
1195  E6AD  88                  DEY
1196  E6AE  10 F6               BPL     DEVS1       ;CONTINUE FOR ALL DEVICES
1197                    ;
1198                    ; NO DEVICE FOUND, DECLARE NON-EXISTENT DEVICE ERROR
1199  E6B0  A0 82       CIERR2: LDY     #NONDEV     ;ERROR CODE
1200  E6B2  38                  SEC                 ;SHOW ERROR
1201  E6B3  B0 13               BCS     DEVS4       ;AND RETURN
1202                    ;
1203                    ; FOUND DEVICE, SET ICHID,ICDNO, AND INIT DEVICE
1204  E6B5  98          DEVS2:  TYA
1205  E6B6  85 20               STA     ICHIDZ      ;SAVE HANDLER INDEX
1206  E6B8  38                  SEC
1207  E6B9  A0 01               LDY     #1
1208  E6BB  B1 24               LDA     (ICBALZ),Y  ;GET DEVICE NUMBER (DRIVE NUMBER)
1209  E6BD  E9 30               SBC     #ASCZER     ;SUBTRACT ASCII ZERO
1210  E6BF  C9 0A               CMP     #$A         ;IS NUMBER IN RANGE?
1211  E6C1  90 02               BCC     DEVS3       ;YES
1212  E6C3  A9 01               LDA     #1          ;NO, DEFAULT TO ONE
1213  E6C5  85 21       DEVS3:  STA     ICDNOZ      ;SAVE DEVICE NUMBER
1214  E6C7  18                  CLC                 ;SHOW NO ERROR
1215                    ;
1216                    ; RETURN
1217  E6C8  60          DEVS4:  RTS
```

```
1218                                    .PAGE
1219                            ;
1220                            ;
1221                            ; CIO ROM TABLES
1222                            ;
1223                            ; COMMAND TABLE
1224                            ; MAPS EACH COMMAND TO OFFSET FOR APPROPRIATE VECTOR IN HANDLER
1225  E6C9  00 04 04 04         COMTAB: .BYTE   0,4,4,4,4,6,6,6,6,2,8,10
1226  E6CD  04 06 06 06
1227  E6D1  06 02 08 0A
1228  022F                      LENGTH=*-CIOINT
1229  E6D5                      CRNTP1  =*
1230                                    *=$14
1231  0014  00                  CIOSPR: .BYTE   INTORG-CRNTP1 ;^GCIOL IS TOO LONG
```

```
1232
1233                                    LIST    S
1234                                    .TITLE  'INTERRUPT  HANDLER'
1235                            ;LIVES ON DK1:INTHV.SRC
1236  0006                      SRTIM2  =       6               ;SECOND REPEAT INTERVAL
1237                                    *=SETVBV
1238  E45C  4C 12 E9                    JMP     SETVBL
1239  E45F  4C D1 E7                    JMP     SYSVBL
1240  E462  4C 3E E9                    JMP     XITVBL
1241                                    *=INTINV
1242  E46B  4C D5 E6                    JMP     IHINIT
1243                            ;
1244                                    *=VCTABL+INTABS-VDSLST
1245                            ;                               ;
1246  E480  B3 E7                       .WORD   SYRTI           ;VDSLST
1247  E482  B2 E7                       .WORD   SYIRQB          ;VPRCED
1248  E484  B2 E7                       .WORD   SYIRQB          ;VINTER
1249  E486  B2 E7                       .WORD   SYIRQB          ;VBREAK
1250                            ;
1251  E488                              .RES    8
1252  E490  B2 E7                       .WORD   SYIRQB          ;VTIMR1
1253  E492  B2 E7                       .WORD   SYIRQB          ;VTIMR2
1254  E494  B2 E7                       .WORD   SYIRQB          ;VTIMR4
1255  E496  F6 E6                       .WORD   SYIRQ           ;VIMIRQ
1256  E498  00 00 00 00                 .WORD   0,0,0,0,0       ;CDTMV1-4
1257  E49C  00 00 00 00
1258  E4A0  00 00
1259  E4A2  D1 E7                       .WORD   SYSVBL          ;VVBLKI
1260  E4A4  3E E9                       .WORD   XITVBL          ;VVBLKD
1261                            ;
1262                                    *=$900C
1263                            ;
1264  900C  A9 E6                       LDA     #PIRQH          ;SET UP RAM VECTORS FOR LINBUG VERSION
1265  900E  8D F9 FF                    STA     $FFF9
1266  9011  A9 F3                       LDA     #PIRQL
1267  9013  8D F8 FF                    STA     $FFF8
1268  9016  A9 E7                       LDA     #PNMIH
1269  9018  8D FB FF                    STA     $FFFB
1270  901B  A9 B4                       LDA     #PNMIL
1271  901D  8D FA FF                    STA     $FFFA
1272  9020  60                          RTS
```

```
1273                            .PAGE
1274                    ;
1275                    ; IRQ HANDLER
1276                    ;
1277                    ; JUMP THRU IMMEDIATE IRQ VECTOR, WHICH ORDINARILY POINTS TO
1278                    ; SYSTEM IRQ: DETERMINE & CLEAR CAUSE, JUMP THRU SOFTWARE VECTOR.
1279                    ;
1280                            *=INTORG
1281 E6D5 A9 40        IHINIT: LDA     #$40            ;VBL ON BUF DLIST OFF***FOR NOW***
1282 E6D7 8D 0E D4              STA     NMIEN           ;ENABLE DISPLAY LIST, VERTICAL BLANK
1283 E6DA A9 38                 LDA     #$38            ;LOOK AT DATA DIRECTION REGISTERS IN PIA
1284 E6DC 8D 02 D3              STA     PACTL
1285 E6DF 8D 03 D3              STA     PBCTL
1286 E6E2 A9 00                 LDA     #0              ;MAKE ALL INPUTS
1287 E6E4 8D 00 D3              STA     PORTA
1288 E6E7 8D 01 D3              STA     PORTB
1289 E6EA A9 3C                 LDA     #$3C            ;BACK TO PORTS
1290 E6EC 8D 02 D3              STA     PACTL
1291 E6EF 8D 03 D3              STA     PBCTL
1292 E6F2 60                    RTS
1293 E6F3 6C 16 02      PIRQ:   JMP     (VIMIRQ)
1294 E6F6 48            SYIRQ:  PHA                     ;SAVE A ON STACK
1295 E6F7 A9 10                 LDA     #$10            ;SERIAL OUT READ BIT
1296 E6F9 2C 0E D2              BIT     IRQST           ;SEE IF SER OUT RDY
1297 E6FC D0 0D                 BNE     SYIRQ1          ;JUMP IF NOT
1298 E6FE A9 EF                 LDA     #$EF
1299 E700 8D 0E D2              STA     IRQEN
1300 E703 A5 10                 LDA     POKMSK          ;RESET ENABLE BYTE
1301 E705 8D 0E D2              STA     IRQEN
1302 E708 6C 0C 02              JMP     (VSEROR)        ;JUMP THRU VECTOR
1303 E70B A9 20         SYIRQ1: LDA     #$20            ;SERIAL IN READY BIT
1304 E70D 2C 0E D2              BIT     IRQST
1305 E710 D0 0D                 BNE     SYIRQ2
1306 E712 A9 DF                 LDA     #$DF
1307 E714 8D 0E D2              STA     IRQEN
1308 E717 A5 10                 LDA     POKMSK
1309 E719 8D 0E D2              STA     IRQEN
1310 E71C 6C 0A 02              JMP     (VSERIN)
1311 E71F A9 08         SYIRQ2: LDA     #$08            ;SERIAL OUT COMPLETE
1312 E721 24 10                 BIT     POKMSK          ;ONLY CHECK IF ENABLED
1313 E723 F0 12                 BEQ     SYIRQ3          ;SINCE THIS IS A WEIRD CASE IN THE
1314 E725 2C 0E D2              BIT     IRQEN           ;HARDWARE
1315 E728 D0 0D                 BNE     SYIRQ3
1316 E72A A9 F7                 LDA     #$F7
1317 E72C 8D 0E D2              STA     IRQEN
1318 E72F A5 10                 LDA     POKMSK
1319 E731 8D 0E D2              STA     IRQEN
1320 E734 6C 0E 02              JMP     (VSEROC)
1321 E737 AD 0E D2      SYIRQ3: LDA     IRQEN           ;TO CHECK TIMERS
1322 E73A 6A                    ROR     A
1323 E73B B0 0D                 BCS     SYIRQ4          ;TIMER1
1324 E73D A9 FE                 LDA     #$FE
1325 E73F 8D 0E D2              STA     IRQEN
1326 E742 A5 10                 LDA     POKMSK
```

```
1327  E744  8D 0E D2          STA   IRQEN
1328  E747  6C 10 02          JMP   (VTIMR1)
1329  E74A  6A          SYIRQ4: ROR   A
1330  E74B  B0 0D              BCS   SYIRQ5
1331  E74D  A9 FD              LDA   #$FD
1332  E74F  8D 0E D2          STA   IRQEN
1333  E752  A5 10              LDA   POKMSK
1334  E754  8D 0E D2          STA   IRQEN
1335  E757  6C 12 02          JMP   (VTIMR2)
1336  E75A  6A          SYIRQ5: ROR   A
1337  E75B  B0 0A              BCS   SYIRQ6
1338  E75D  A9 FB              LDA   #$FB
1339  E75F  8D 0E D2          STA   IRQEN
1340  E762  A5 10              LDA   POKMSK
1341  E764  8D 0E D2          STA   IRQEN
1342                     ;     JMP   (VTIMR4)
1343  E767  2C 0E D2     SYIRQ6: BIT   IRQEN       ;CHECK BREAK KEY, KEYBOARD INTERRUPTS
1344  E76A  70 0D              BVS   SYIRQ7       ;IF KEYBOARD
1345  E76C  A9 BF              LDA   #$BF
1346  E76E  8D 0E D2          STA   IRQEN
1347  E771  A5 10              LDA   POKMSK
1348  E773  8D 0E D2          STA   IRQEN
1349  E776  6C 08 02          JMP   (VKEYBD)
1350  E779  30 18        SYIRQ7: BMI   SYIRQ8       ;BREAK KEY
1351  E77B  A9 7F              LDA   #$7F
1352  E77D  8D 0E D2          STA   IRQEN
1353  E780  A5 10              LDA   POKMSK
1354  E782  8D 0E D2          STA   IRQEN
1355  E785  A9 00              LDA   #0
1356  E787  85 11              STA   BRKKEY       ;SET BREAK KEY FLAG
1357  E789  8D FF 02           STA   SSFLAG       ;START/STOP FLAG
1358  E78C  8D F0 02           STA   CRSINH       ;CURSOR INHIBIT
1359  E78F  85 4D              STA   ATRACT       ;TURN OFF ATRACT MODE
1360  E791  68                 PLA
1361  E792  40                 RTI                ;EXIT FROM INT
1362  E793  2C 02 D3     SYIRQ8: BIT   PACTL       ;PROCEED ***I GUESS***
1363  E796  10 06              BPL   SYIRQ9
1364  E798  AD 00 D3           LDA   PORTA        ;CLEAR INT STATUS BIT
1365  E79B  6C 02 02           JMP   (VPRCED)
1366  E79E  2C 03 D3     SYIRQ9: BIT   PBCTL       ;INTERRUPT ***I GUESS***
1367  E7A1  10 06              BPL   SYIRQA
1368  E7A3  AD 01 D3           LDA   PORTB        ;CLEAR INT STATUS
1369  E7A6  6C 04 02           JMP   (VINTER)
1370  E7A9  08           SYIRQA: PHP
1371  E7AA  68                 PLA                ;SEE IF SOFTWARE BREAK
1372  E7AB  29 10              AND   #$10         ;B BIT OF P REGISTER
1373  E7AD  F0 03              BEQ   SYIRQB
1374  E7AF  6C 06 02           JMP   (VBREAK)
1375  E7B2  68           SYIRQB: PLA
1376  E7B3  40           SYRTI:  RTI                ;UNIDENTIFIED INTERRUPT, JUST RETURN.
```

```
1377                            .PAGE
1378                    ;
1379                    ; NMI HANDLER
1380                    ;
1381                    ; DETERMINE CAUSE AND JUMP THRU VECTOR
1382                    ;
1383 E7B4 2C 0F D4     PNMI:   BIT     NMIST
1384 E7B7 10 03                BPL     PNMI1           ;SEE IF DISPLAY LIST
1385 E7B9 6C 00 02              JMP     (VDSLST)
1386 E7BC 48           PNMI1:  PHA
1387 E7BD AD 0F D4              LDA     NMIST
1388 E7C0 29 20                 AND     #$20            ;SEE IF RESET
1389 E7C2 F0 03                 BEQ     *+5
1390 E7C4 4C 74 E4              JMP     WARMSV          ;GO THRU WARM START JUMP
1391 E7C7 8A                    TXA                     ;SAVE REGISTERS
1392 E7C8 48                    PHA
1393 E7C9 98                    TYA
1394 E7CA 48                    PHA
1395 E7CB 8D 0F D4              STA     NMIRES          ;RESET INTERRUPT STATUS
1396 E7CE 6C 22 02              JMP     (VVBLKI)        ;JUMP THRU VECTOR
```

```
1397                            .PAGE
1398                    ;
1399                    ; SYSTEM VBLANK ROUTINE
1400                    ;
1401                    ; INC FRAME COUNTER. PROCESS COUNTDOWN TIMERS. EXIT IF I WAS SET, CLEAR
1402                    ; SET DLISTL, DLISTH, DMACTL FROM RAM CELLS. DO SOFTWARE REPEAT.
1403                    ;
1404 E7D1 E6 14        SYSVBL: INC     RTCLOK+2        ;INC FRAME COUNTER
1405 E7D3 D0 08                BNE     SYSVB1
1406 E7D5 E6 4D                INC     ATRACT          ;INCREMENT ATRACT (CAUSES ATRACT WHEN MINUS)
1407 E7D7 E6 13                INC     RTCLOK+1
1408 E7D9 D0 02                BNE     SYSVB1
1409 E7DB E6 12                INC     RTCLOK
1410 E7DD A9 FE        SYSVB1: LDA     #$FE            ;[ATRACT] SET DARK MASK TO NORMAL
1411 E7DF A2 00                LDX     #0              ;SET COLRSH TO NORMAL
1412 E7E1 A4 4D                LDY     ATRACT          ;TEST ATRACT FOR NEGATIVE
1413 E7E3 10 06                BPL     VBATRA          ;WHILE POSITIVE, DONT GO INTO ATRACT
1414 E7E5 85 4D                STA     ATRACT          ;IN ATRACT, SO STAY BY STA $FE
1415 E7E7 A6 13                LDX     RTCLOK+1        ;COLOR SHIFT FOLLOWS RTCLOK+1
1416 E7E9 A9 F6                LDA     #$F6            ;SET DARK MASK TO DARK
1417 E7EB 85 4E        VBATRA: STA     DRKMSK
1418 E7ED 86 4F                STX     COLRSH
1419 E7EF A2 00                LDX     #0              ;POINT TO TIMER1
1420 E7F1 20 F5 E8             JSR     DCTIMR          ;GO DECREMENT TIMER1
1421 E7F4 D0 03                BNE     SYSVB2          ;BRANCH IF STILL COUNTING
1422 E7F6 20 EF E8             JSR     JTIMR1          ;GO JUMP TO ROUTINE
1423 E7F9 A5 42        SYSVB2: LDA     CRITIC
1424 E7FB D0 08                BNE     XXIT            ;GO IF CRITICAL SET
1425 E7FD BA                   TSX                     ;SEE IF I WAS SET
1426 E7FE BD 04 01             LDA     $104,X          ;GET STACKED P
1427 E801 29 04                AND     #$04            ;I BIT
1428 E803 F0 03                BEQ     SYSVB3          ;BRANCH IF OK
1429 E805 4C 3E E9     XXIT:   JMP     XITVBL          ;I WAS SET, EXIT
1430 E808 58           SYSVB3: CLI
1431 E809 AD 0D D4             LDA     PENV
1432 E80C 8D 35 02             STA     LPENV
1433 E80F AD 0C D4             LDA     PENH
1434 E812 8D 34 02             STA     LPENH
1435 E815 AD 31 02             LDA     SDLSTH
1436 E818 8D 03 D4             STA     DLISTH
1437 E81B AD 30 02             LDA     SDLSTL
1438 E81E 8D 02 D4             STA     DLISTL
1439 E821 AD 2F 02             LDA     SDMCTL
1440 E824 8D 00 D4             STA     DMACTL
1441 E827 AD 6F 02             LDA     GPRIOR          ;GLOBAL PRIOR
1442 E82A 8D 1B D0             STA     PRIOR
1443 E82D A9 08                LDA     #$08            ;TURN OFF KEYBOARD SPEAKER
1444 E82F 8D 1F D0             STA     CONSOL
1445 E832 A2 08                LDX     #8
1446 E834 BD C0 02     SCOLLP: LDA     PCOLR0,X        ;LOAD COLOR REGISTERS FROM RAM
1447 E837 45 4F                EOR     COLRSH          ;DO COLOR SHIFT
1448 E839 25 4E                AND     DRKMSK          ;AND DARK ATRACT
1449 E83B 9D 12 D0             STA     COLPM0,X
1450 E83E CA                   DEX
```

```
1451  E83F  10 F3          BPL     SCOLLP
1452  E841  AD F4 02       LDA     CHBAS
1453  E844  8D 09 D4       STA     CHBASE
1454  E847  AD F3 02       LDA     CHACT
1455  E84A  8D 01 D4       STA     CHACTL
1456  E84D  A2 02          LDX     #2              ;POINT TO TIMER 2
1457  E84F  20 F5 E8       JSR     DCTIMR
1458  E852  D0 03          BNE     SYSVB4          ;IF DIDNT GO ZERO
1459  E854  20 F2 E8       JSR     JTIMR2          ;GO JUMP TO TIMER2 ROUTINE
1460  E857  A2 02       SYSVB4: LDX     #2              ;RESTORE X
1461  E859  E8          SYSVBB: INX
1462  E85A  E8                  INX
1463  E85B  BD 18 02       LDA     CDTMV1,X
1464  E85E  1D 19 02       ORA     CDTMV1+1,X
1465  E861  F0 06          BEQ     SYSVBA
1466  E863  20 F5 E8       JSR     DCTIMR          ;DECREMENT AND SET FLAG IF NONZERO
1467  E866  9D 26 02       STA     CDTMF3-4,X
1468  E869  E0 08       SYSVBA: CPX     #8              ;SEE IF DONE ALL 3
1469  E86B  D0 EC          BNE     SYSVBB          ;LOOP
1470                    ;CHECK DEBOUNCE COUNTER
1471  E86D  AD 0F D2       LDA     SKSTAT
1472  E870  29 04          AND     #$04            ;KEY DOWN BIT
1473  E872  F0 08          BEQ     SYVB6A          ;IF KEY DOWN
1474                    ;KEY UP SO COUNT IT
1475  E874  AD F1 02       LDA     KEYDEL          ;KEY DELAY COUNTER
1476  E877  F0 03          BEQ     SYVB6A          ;IF COUNTED DOWN ALREADY
1477  E879  CE F1 02       DEC     KEYDEL          ;COUNT IT
1478                    ; CHECK SOFTWARE REPEAT TIMER
1479  E87C  AD 2B 02    SYVB6A: LDA     SRTIMR
1480  E87F  F0 17          BEQ     SYSVB7          ;DOESNT COUNT
1481  E881  AD 0F D2       LDA     SKSTAT
1482  E884  29 04          AND     #$04            ;CHECK KEY DOWN BIT
1483  E886  D0 60          BNE     SYSVB6          ;BRANCH IF NO LONGER DOWN
1484  E888  CE 2B 02       DEC     SRTIMR          ;COUNT FRAME OF KEY DOWN
1485  E88B  D0 0B          BNE     SYSVB7          ;BRANCH IF NOT RUN OUT
1486                    ; TIMER RAN OUT - RESET AND SIMULATE KEYBOARD IRQ
1487  E88D  A9 06          LDA     #SRTIM2         ;TIMER VALUE
1488  E88F  8D 2B 02       STA     SRTIMR          ;SET TIMER
1489  E892  AD 09 D2       LDA     KBCODE          ;GET THE KEY
1490  E895  8D FC 02       STA     CH              ;PUT INTO CH
1491                    ; READ GAME CONTROLLERS
1492  E898  A0 01       SYSVB7: LDY     #1
1493  E89A  A2 03          LDX     #3
1494  E89C  B9 00 D3    STLOOP: LDA     PORTA,Y
1495  E89F  4A                  LSR     A
1496  E8A0  4A                  LSR     A
1497  E8A1  4A                  LSR     A
1498  E8A2  4A                  LSR     A
1499  E8A3  9D 78 02       STA     STICKO,X        ;STORE JOYSTICK
1500  E8A6  CA                  DEX
1501  E8A7  B9 00 D3       LDA     PORTA,Y
1502  E8AA  29 0F          AND     #$F
1503  E8AC  9D 78 02       STA     STICKO,X        ;STORE JOYSTICK
1504  E8AF  CA                  DEX
```

```
1505 E8B0 88          DEY
1506 E8B1 10 E9       BPL     STLOOP
1507                  ;
1508 E8B3 A2 03       LDX     #3
1509 E8B5 BD 10 D0    STRL:   LDA     TRIG0,X     ;MOVE JOYSTICK TRIGGERS
1510 E8B8 9D 84 02            STA     STRIG0,X
1511 E8BB BD 00 D2            LDA     POT0,X      ;MOVE POT VALUES
1512 E8BE 9D 70 02            STA     PADDL0,X
1513 E8C1 BD 04 D2            LDA     POT4,X
1514 E8C4 9D 74 02            STA     PADDL4,X
1515 E8C7 CA                  DEX
1516 E8C8 10 EB               BPL     STRL
1517 E8CA 8D 0B D2            STA     POTGO       ;START POTS FOR NEXT TIME
1518                  ;
1519 E8CD A2 06       LDX     #6
1520 E8CF A0 03       LDY     #3
1521 E8D1 B9 78 02    PTRLP:  LDA     STICK0,Y    ;TRANSFER BITS FROM JOYSTICKS
1522 E8D4 4A                  LSR     A           ;TO PADDLE TRIGGERS
1523 E8D5 4A                  LSR     A
1524 E8D6 4A                  LSR     A
1525 E8D7 9D 7D 02            STA     PTRIG1,X
1526 E8DA A9 00               LDA     #0
1527 E8DC 2A                  ROL     A
1528 E8DD 9D 7C 02            STA     PTRIG0,X
1529 E8E0 CA                  DEX
1530 E8E1 CA                  DEX
1531 E8E2 88                  DEY
1532 E8E3 10 EC               BPL     PTRLP
1533                  ;
1534 E8E5 6C 24 02            JMP     (VVBLKD)    ;GO TO DEFERRED VBLANK ROUTINE
1535 00E8             SV7H    =       SYSVB7/256
1536 0098             SV7L    =       (-256)*SV7H+SYSVB7
1537 E8E8 A9 00       SYSVB6: LDA     #0
1538 E8EA 8D 2B 02            STA     SRTIMR      ;ZERO TIMER
1539 E8ED F0 A9               BEQ     SYSVB7      ;UNCOND
1540 E8EF 6C 26 02    JTIMR1: JMP     (CDTMA1)
1541 E8F2 6C 28 02    JTIMR2: JMP     (CDTMA2)
1542                  ;
1543                  ; SUBROUTINE TO DECREMENT A COUNTDOWN TIMER
1544                  ; ENTRY X=OFFSET FROM TIMER 1
1545                  ; EXIT A,P=ZERO IF WENT ZERO, FF OTHERWISE
1546                  ;
1547 E8F5 BC 18 02    DCTIMR: LDY     CDTMV1,X    ;LO BYTE
1548 E8F8 D0 08               BNE     DCTIM1      ;NONZERO, GO DEC IT
1549 E8FA BC 19 02            LDY     CDTMV1+1,X  ;SEE IF BOTH ZERO
1550 E8FD F0 10               BEQ     DCTXF       ;YES, EXIT NONZERO
1551 E8FF DE 19 02            DEC     CDTMV1+1,X  ;DEC HI BYTE
1552 E902 DE 18 02    DCTIM1: DEC     CDTMV1,X    ;DEC LO BYTE
1553 E905 D0 08               BNE     DCTXF
1554 E907 BC 19 02            LDY     CDTMV1+1,X
1555 E90A D0 03               BNE     DCTXF
1556 E90C A9 00               LDA     #0          ;WENT ZERO, RETURN ZERO
1557 E90E 60                  RTS
1558 E90F A9 FF       DCTXF:  LDA     #$FF        ;RETURN NONZERO
```

```
    1559 E911 60          RTS
```

```
1560                                    PAGE
1561                            ;
1562                            ; SUBROUTINE TO SET VERTICAL BLANK VECTORS AND TIMERS
1563                            ; ENTRY X=HI, Y=LO BYTE TO SET
1564                            ;       A= 1-5 TIMERS 1-5
1565                            ;          6 IMM VBLANK
1566                            ;          7 DEF VBLANK
1567                            ;
1568 E912 0A                   SETVBL: ASL     A               ;MUL BY 2
1569 E913 8D 2D 02                     STA     INTEMP
1570 E916 A9 00                        LDA     #$00            ;DLIST OFF BUT VBLANK OFF***FOR NOW***
1571 E918 8D 0E D4                     STA     NMIEN
1572 E91B 8A                           TXA
1573 E91C AE 2D 02                     LDX     INTEMP
1574 E91F 9D 17 02                     STA     CDTMV1-1,X
1575 E922 98                           TYA
1576 E923 9D 16 02                     STA     CDTMV1-2,X
1577 E926 A9 40                        LDA     #$40            ;DLIST OFF***FOR NOW***, VBLANK ON
1578 E928 8D 0E D4                     STA     NMIEN
1579 E92B 2C 0F D4                     BIT     NMIST           ;SEE IF A VBLANK HAPPENED
1580 E92E 50 0D                        BVC     SETVB1          ;IF NOT, GO EXIT
1581                            ; SIMULATE VERTICAL BLANK NMI
1582 E930 A9 E9                        LDA     #SV1H
1583 E932 48                           PHA                     ;PUSH PC ON STACK AS IF INTERRUPT
1584 E933 A9 3D                        LDA     #SV1L
1585 E935 48                           PHA
1586 E936 08                           PHP                     ;DUMMY P
1587 E937 48                           PHA                     ;DUMMY A
1588 E938 48                           PHA                     ;DUMMY X
1589 E939 48                           PHA                     ;DUMMY Y
1590 E93A 6C 22 02                     JMP     (VVBLKI)        ;GO THRU VBLANK VECTOR
1591 E93D 60                   SETVB1: RTS
1592 00E9                      SV1H    =       SETVB1/256
1593 003D                      SV1L    =       (-256)*SV1H+SETVB1
1594                            ;
1595                            ; EXIT FROM VERTICAL BLANK
1596                            ;
1597 E93E 68                   XITVBL: PLA                     ;UNSTACK Y
1598 E93F A8                           TAY
1599 E940 68                           PLA                     ;UNSTACK X
1600 E941 AA                           TAX
1601 E942 68                           PLA                     ;UNSTACK A
1602 E943 40                           RTI                     ;AND GO BACK FROM WHENCE.
1603 00E6                      PIRQH   =       PIRQ/256
1604 00F3                      PIRQL   =       (-256)*PIRQH+PIRQ
1605 00E7                      PNMIH   =       PNMI/256
1606 00B4                      PNMIL   =       (-256)*PNMIH+PNMI
1607                           ;SPARE BYTE OR MODULE TOO LONG FLAG
1608 E944                      CRNTP2  =*
1609                                   *=$14
1610 0014 00                   INTSPR: .BYTE   SIOORG-CRNTP2 ;^GINTHV IS TOO LONG
```

```
1611
1612                            .TITLE  'SIO ( SERIAL BUS INPUT/OUTPUT CONTROLLER )'
1613                    ;       COLLEEN OPERATING SYSTEM
1614                    ;
1615                    ;       SIO  ( SERIAL BUS INPUT/OUTPUT CONTROLLER )
1616                    ;       WITH SOFTWARE BAUD RATE CORRECTION ON CASSETTE
1617                    ;
1618                    ;
1619                    ;               AL MILLER       3-APR-79
1620                    ;
1621                    ;
1622                    ;THIS MODULE HAS ONE ENTRY POINT.  IT IS CALLED BY THE DEVICE
1623                    ;HANDLERS.  IT INTERPRETS A PREVIOUSLY ESTABLISHED DEVICE CONTROL
1624                    ;BLOCK (STORED IN GLOBAL RAM) TO ISSUE COMMANDS
1625                    ;TO THE SERIAL BUS TO CONTROL TRANSMITTING AND RECEIVING DATA.
1626                    ;
1627                    ;
1628                    ;
1629                    ;
```

```
1630                            .PAGE
1631                    ;EQUATES
1632                    ;
1633                    ;DCD DEVICE BUS ID NUMBERS
1634  0030              FLOPPY  =       $30
1635                    ;PRINTR =        $40
1636                    ;CASSET =        $60            ;!!!!! *****
1637  0060              CASET   =       $60             ;!!!!! *****
1638                    ;
1639                    ;
1640                    ;BUS COMMANDS
1641                    ;
1642  0052              READ    =        'R
1643  0057              WRITE   =        'W
1644                    ;STATIS =        'S
1645                    ;FORMAT =        '!
1646                    ;
1647                    ;
1648                    ;COMMAND AUX BYTES
1649                    ;
1650  0053              SIDWAY  =        'S             ;PRINT 16 CHARACTERS SIDEWAYS
1651  004E              NORMAL  =        'N             ;PRINT 40 CHARACTERS NORMALLY
1652  0044              DOUBLE  =        'D             ;PRINT 20 CHARACTERS DOUBLE WIDE
1653  0050              PLOT    =        'P             ;PLOT MODE
1654                    ;
1655                    ;
1656                    ;BUS RESPONSES
1657                    ;
1658  0041              ACK     =        'A             ;DEVICE ACKNOWLEDGES INFORMATION
1659  004E              NACK    =        'N             ;DEVICE DID NOT UNDERSTAND
1660  0043              COMPLT  =        'C             ;DEVICE SUCCESSFULLY COMPLETED OPERATION
1661  0045              ERROR   =        'E             ;DEVICE INCURRED AN ERROR IN AN ATTEMPTED
1662                    ;
1663                    ;
1664                    ;MISCELLANEOUS EQUATES
1665                    ;
1666  0028              B192LO  =       $28            ;19200 BAUD RATE POKEY COUNTER VALUES (LO BY
1667  0000              B192HI  =       $00            ;(HI BYTE)
1668  00CC              B600LO  =       $CC            ;600 BAUD (LO BYTE)
1669  0005              B600HI  =       $05            ;(HI BYTE)
1670  0005              HITONE  =       $05            ;FSK HI FREQ POKEY COUNTE VALUE (5326 HZ)
1671  0007              LOTONE  =       $07            ;FSK LO FREQ POKEY COUNTER VALUE (3995 HZ)
1672                    ;
1673  00B4              WIRGLO  =       180            ;WRITE INTER RECORD GAP (IN 1/60 SEC)
1674  0000              WIRGHI  =       0
1675  0078              RIRGLO  =       120            ;READ INTER RECORD GAP (IN 1/60 SEC)
1676  0000              RIRGHI  =       0
1677  000F              WSIRG   =       15             ;SHORT WRITE INTER RECORD GAP
1678  000A              RSIRG   =       10             ;SHORT READ INTER RECORD GAP
1679                    ;
1680  0034              NCOMLO  =       $34            ;PIA COMMAND TO LOWER NOT COMMAND LINE
1681  003C              NCOMHI  =       $3C            ;PIA COMMAND TO RAISE NOT COMMAND LINE
1682  0034              MOTRGO  =       $34            ;PIA COMMAND TO TURN ON CASSETTE MOTOR
1683  003C              MOTRST  =       $3C            ;PIA COMMAND TO TURN OFF MOTOR
```

```
1684                    ;
1685  0002              TEMPHI  =       TEMP/256    ;ADDRESS OF TEMP CELL (HI BYTE)
1686  003E              TEMPLO  =       (-256)*TEMPHI+TEMP ;(LO BYTE)
1687  0002              CBUFHI  =       CDEVIC/256  ;ADDRESS OF COMMAND BUFFER (HI BYTE)
1688  003A              CBUFLO  =       (-256)*CBUFHI+CDEVIC ;(LO BYTE)
1689                    ;
1690  000D              CRETRI  =       13          ;NUMBER OF COMMAND FRAME RETRIES
1691  0001              DRETRI  =       1           ;NUMBER OF DEVICE RETRIES
1692  0002              CTIMLO  =       2           ;COMMAND FRAME ACK TIME OUT (LO BYTE)
1693  0000              CTIMHI  =       0           ;COMMAND FRAME ACK TIME OUT (HI BYTE)
1694                    ;
1695                    ;
1696                    ;JTADRH  =       JTIMER/256 HI BYTE OF JUMP TIMER ROUTINE ADDR
1697                    ;                MOVED TO LINE 1427
1698                    ;JTADRL  =       (-256)*JTADRH+JTIMER    "MOVED TO LINE 1428"
1699                    ;
```

```
1700                                    .PAGE
1701                            ;       SIO
1702                            ;
1703                            ;
1704                                    *=SIOV
1705  E459  4C 59 E9                   JMP     SIO             ;SIO ENTRY POINT
1706                            ;
1707                                    *=SIOINV
1708  E465  4C 44 E9                   JMP     SIOINT          ;SIO INITIALIZATION ENTRY POINT
1709                            ;
1710                                    *=SENDEV
1711  E468  4C F6 EB                   JMP     SENDEN          ;SEND ENABLE ENTRY POINT
1712                            ;
1713                                    *=VCTABL-INTABS+VSERIN
1714                            ;
1715  E48A  11 EB                       .WORD   ISRSIR          ;VSERIN
1716  E48C  90 EA                       .WORD   ISRODN          ;VSEROR
1717  E48E  D1 EA                       .WORD   ISRTD           ;VSEROC
1718                            ;
1719                            ;
1720                            ;
1721                                    *=SIOORG
1722                            ;
1723                            ;SIO INITIALIZATION SUBROUTINE
1724                            ;
1725  E944  A9 3C               SIOINT: LDA     #MOTRST
1726  E946  8D 02 D3                    STA     PACTL           ;TURN OFF MOTOR
1727                            ;
1728  E949  A9 3C                       LDA     #NCOMHI
1729  E94B  8D 03 D3                    STA     PBCTL           ;RAISE NOT COMMAND LINE
1730                            ;
1731                            ;
1732  E94E  A9 03                       LDA     #3
1733  E950  8D 32 02                    STA     SSKCTL          ;GET POKEY OUT OF INITIALIZE MODE
1734  E953  85 41                       STA     SOUNDR          ;INIT POKE ADDRESS FOR QUIET I/O
1735  E955  8D 0F D2                    STA     SKCTL
1736                            ;
1737                            ;
1738  E958  60                          RTS                     ;RETURN
1739                            ;
1740                            ;
1741                            ;
1742                            ;
1743                            ;
1744                            ;
1745  E959  BA                  SIO:    TSX
1746  E95A  8E 18 03                    STX     STACKP          ;SAVE STACK POINTER
1747  E95D  A9 01                       LDA     #1
1748  E95F  85 42                       STA     CRITIC
1749                            ;
1750  E961  AD 00 03                    LDA     DDEVIC
1751  E964  C9 60                       CMP     #CASET
1752  E966  D0 03                       BNE     NOTCST          ;BRANCH IF NOT CASSETTE
1753  E968  4C 84 EB                   JMP     CASENT          ;OTHERWISE JUMP TO CASSETTE ENTER
```

```
1754                            ;
1755                            ;ALL DEVICES EXCEPT CASSETTE ARE INTELLIGENT
1756                            ;
1757  E96B  A9 00       NOTCST: LDA     #0
1758  E96D  8D 0F 03             STA     CASFLG      ;INIT CASSETTE FLAG TO NO CASSETTE
1759                            ;
1760  E970  A9 01               LDA     #DRETRI     ;SET NUMBER OF DEVICE RETRIES
1761  E972  85 37               STA     DRETRY
1762  E974  A9 0D       COMMND: LDA     #CRETRI     ;SET NUMBER OF COMMAND FRAME RETRIES
1763  E976  85 36               STA     CRETRY
1764                            ;
1765                            ;SEND A COMMAND FRAME
1766                            ;
1767  E978  A9 28       COMFRM: LDA     #B192LO     ;SET BAUD RATE TO 19200
1768  E97A  8D 04 D2            STA     AUDF3
1769  E97D  A9 00               LDA     #B192HI
1770  E97F  8D 06 D2            STA     AUDF4
1771                            ;
1772  E982  18                  CLC                 ;SET UP COMMAND BUFFER
1773  E983  AD 00 03            LDA     DDEVIC
1774  E986  6D 01 03            ADC     DUNIT
1775  E989  69 FF               ADC     #$FF        ;SUBTRACT 1
1776  E98B  8D 3A 02            STA     CDEVIC      ;SET BUS ID NUMBER
1777                            ;
1778  E98E  AD 02 03            LDA     DCOMND
1779  E991  8D 3B 02            STA     CCOMND      ;SET BUS COMMAND
1780                            ;
1781  E994  AD 0A 03            LDA     DAUX1       ;STORE COMMAND FRAME AUX BYTES 1 AND 2
1782  E997  8D 3C 02            STA     CAUX1
1783  E99A  AD 0B 03            LDA     DAUX2
1784  E99D  8D 3D 02            STA     CAUX2       ;DONE SETTING UP COMMAND BUFFER
1785                            ;
1786  E9A0  18                  CLC                 ;SET BUFFER POINTER TO COMMAND FRAME BUFFER
1787  E9A1  A9 3A               LDA     #CBUFLO
1788  E9A3  85 32               STA     BUFRLO      ;AND BUFFER END ADDRESS
1789  E9A5  69 04               ADC     #4
1790  E9A7  85 34               STA     BFENLO
1791  E9A9  A9 02               LDA     #CBUFHI
1792  E9AB  85 33               STA     BUFRHI
1793  E9AD  85 35               STA     BFENHI      ;DONE SETTING UP BUFFER POINTER
1794                            ;
1795  E9AF  A9 34               LDA     #NCOMLO     ;LOWER NOT COMMAND LINE
1796  E9B1  8D 03 D3            STA     PBCTL
1797                            ;
1798  E9B4  20 8E EC            JSR     SENDIN      ;SEND THE COMMAND FRAME TO A SMART DEVICE
1799                            ;
1800  E9B7  AD 3F 02            LDA     ERRFLG
1801  E9BA  D0 03               BNE     BADCOM      ;BRANCH IF AN ERROR RECEIVED
1802                            ;
1803  E9BC  98                  TYA
1804  E9BD  D0 07               BNE     ACKREC      ;BRANCH IF ACK RECEIVED
1805                            ;
1806                            ;
1807  E9BF  C6 36       BADCOM: DEC     CRETRY      ;A NACK OR TIME OUT OCCURED
```

```
1808  E9C1  10 B5              BPL     COMFRM      ;SO BRANCH IF ANY RETRIES LEFT
1809                          ;
1810  E9C3  4C 06 EA          JMP     DERR1       ;OTHERWISE, JUMP TO RETURN SECTION
1811                          ;
1812                          ;
1813  E9C6  AD 03 03          ACKREC: LDA     DSTATS      ;ACK WAS RECEIVED
1814  E9C9  10 0C              BPL     WATCOM      ;BRANCH TO WAIT FOR COMPLETE ,
1815                          ; IF THERE IS NO DATA TO BE SENT
1816                          ;
1817                          ;
1818                          ;
1819                          ;SEND A DATA FRAME TO PERIPHERAL
1820                          ;
1821  E9CB  A9 0D              LDA     #CRETRI     ;SET NUMBER OF RETRIES
1822  E9CD  85 36              STA     CRETRY
1823                          ;
1824  E9CF  20 6E EB          JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
1825                          ;
1826  E9D2  20 8E EC          JSR     SENDIN      ;GO SEND THE DATA FRAME TO A SMART DEVICE
1827                          ;
1828  E9D5  F0 E8              BEQ     BADCOM      ;BRANCH IF BAD
1829                          ;
1830                          ;
1831                          ;
1832                          ;WAIT FOR COMPLETE SIGNAL FROM PERIPHERAL
1833                          ;
1834  E9D7  20 79 EC          WATCOM: JSR     STTMOT      ;SET DDEVICE TIME OUT VALUES IN Y,X
1835                          ;
1836  E9DA  A9 00              LDA     #$00
1837  E9DC  8D 3F 02          STA     ERRFLG      ;CLEAR ERROR FLAG
1838                          ;
1839  E9DF  20 9F EC          JSR     WAITER      ;SET UP TIMER AND WAIT
1840  E9E2  F0 12              BEQ     DERR        ;BRANCH IF TIME OUT
1841                          ;
1842                          ;
1843                          ;DEVICE DID NOT TIME OUT
1844                          ;
1845  E9E4  2C 03 03          BIT     DSTATS
1846  E9E7  70 07              BVS     MODATA      ;BRANCH IF MORE DATA FOLLOWS
1847                          ;
1848  E9E9  AD 3F 02          LDA     ERRFLG
1849  E9EC  D0 18              BNE     DERR1       ;BRANCH IF AN ERROR OCCURRED
1850  E9EE  F0 1D              BEQ     RETURN      ;OTHERWISE RETURN
1851                          ;
1852                          ;
1853                          ;
1854                          ;
1855                          ;RECEIVE A DATA FRAME FROM PERIPHERAL
1856                          ;
1857  E9F0  20 6E EB          MODATA: JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
1858                          ;
1859  E9F3  20 E2 EA          JSR     RECEIV      ;GO RECEIVE A DATA FRAME
1860                          ;
1861  E9F6  AD 3F 02          DERR:   LDA     ERRFLG
```

```
1862  E9F9  F0 05          BEQ     NOTERR      ;BRANCH IF NO ERROR PRECEEDED DATA
1863                     ;
1864  E9FB  AD 19 03       LDA     TSTAT       ;GET TEMP STATUS
1865  E9FE  85 30          STA     STATUS      ;STORE IN REAL STATUS
1866                     ;
1867                     ;
1868  EA00  A5 30       NOTERR: LDA     STATUS
1869  EA02  C9 01             CMP     #SUCCES
1870  EA04  F0 07             BEQ     RETURN      ;BRANCH IF COMPLETELY SUCCESSFUL
1871                     ;
1872  EA06  C6 37       DERR1:  DEC     DRETRY
1873  EA08  30 03             BMI     RETURN      ;BRANCH IF OUT OF DEVICE RETRIES
1874                     ;
1875  EA0A  4C 74 E9          JMP     COMMND      ;OTHERWISE, ONE MORE TIME
1876                     ;
1877                     ;
1878                     ;
1879                     ;
1880  EA0D  20 63 EC    RETURN: JSR     SENDDS      ;DISABLE POKEY INTERRUPTS
1881  EA10  A9 00             LDA     #0
1882  EA12  85 42             STA     CRITIC
1883  EA14  A4 30             LDY     STATUS      ;RETURN STATUS IN Y
1884  EA16  8C 03 03          STY     DSTATS      ;AND THE DCB STATUS WORD
1885  EA19  60                RTS     RETURN
1886                     ;
1887                     ;
1888                     ;
1889                     ;
1890                     ;WAIT SUBROUTINE
1891                     ;
1892                     ;WAITS FOR COMPLETE OR ACK
1893                     ;RETURNS Y=$FF IF SUCCESSFUL, Y=$00 IF NOT
1894                     ;
1895  EA1A  A9 00       WAIT:   LDA     #$00
1896  EA1C  8D 3F 02          STA     ERRFLG      ;CLEAR ERROR FLAG
1897                     ;
1898  EA1F  18                CLC                 ;LOAD BUFFER POINTER WITH ADDRESS
1899  EA20  A9 3E             LDA     #TEMPLO     ;OF TEMPORARY RAM CELL
1900  EA22  85 32             STA     BUFRLO
1901  EA24  69 01             ADC     #1
1902  EA26  85 34             STA     BFENLO      ;ALSO SET BUFFER END +1 ADDRESS
1903  EA28  A9 02             LDA     #TEMPHI
1904  EA2A  85 33             STA     BUFRHI
1905  EA2C  85 35             STA     BFENHI      ;DONE LOADING POINTER
1906                     ;
1907  EA2E  A9 FF             LDA     #$FF
1908  EA30  85 3C             STA     NOCKSM      ;SET NO CHECKSUM FOLLOWS DATA FLAG
1909                     ;
1910  EA32  20 E2 EA          JSR     RECEIV      ;GO RECEIVE A BYTE
1911                     ;
1912  EA35  A0 FF             LDY     #$FF        ;ASSUME SUCCESS
1913  EA37  A5 30             LDA     STATUS
1914  EA39  C9 01             CMP     #SUCCES
1915  EA3B  D0 19             BNE     NWOK        ;BRANCH IF IT DID NOT WORK OK
```

```
1916                                ;
1917                                ;
1918                                ;
1919                                ;
1920  EA3D  AD 3E 02          WOK:  LDA   TEMP        ;MAKE SURE THE BYTE SUCCESSFULLY RECEIVED
1921  EA40  C9 41                   CMP   #ACK        ;WAS ACTUALLY AN ACK OR COMPLETE
1922  EA42  F0 21                   BEQ   GOOD
1923  EA44  C9 43                   CMP   #COMPLT
1924  EA46  F0 1D                   BEQ   GOOD
1925                                ;
1926  EA48  C9 45                   CMP   #ERROR
1927  EA4A  D0 06                   BNE   NOTDER      ;BRANCH IF DEVICE DID NOT SEND BACK
1928                                ;A DEVICE ERROR CODE
1929  EA4C  A9 90                   LDA   #DERROR
1930  EA4E  85 30                   STA   STATUS      ;SET DEVICE ERROR STATUS
1931  EA50  D0 04                   BNE   NWOK
1932                                ;
1933  EA52  A9 8B             NOTDER: LDA   #DNACK    ;OTHERWISE SET NACK STATUS
1934  EA54  85 30                   STA   STATUS
1935                                ;
1936  EA56  A5 30             NWOK: LDA   STATUS
1937  EA58  C9 8A                   CMP   #TIMOUT
1938  EA5A  F0 07                   BEQ   BAD         ;BRANCH IF TIME OUT
1939                                ;
1940  EA5C  A9 FF                   LDA   #$FF
1941  EA5E  8D 3F 02                STA   ERRFLG      ;SET SOME ERROR FLAG
1942  EA61  D0 02                   BNE   GOOD        ;RETURN WITH OUT SETTING Y = 0
1943                                ;
1944  EA63  A0 00             BAD:  LDY   #0
1945                                ;
1946  EA65  A5 30             GOOD: LDA   STATUS
1947  EA67  8D 19 03                STA   TSTAT
1948  EA6A  60                      RTS               ;RETURN
1949                                ;
1950                                ;
1951                                ;
1952                                ;
1953                                ;
1954                                ;SEND SUBROUTINE
1955                                ;
1956                                ;SENDS A BUFFER OF BYTES OUT OVER THE SERIAL BUS
1957                                ;
1958                                ;
1959  EA6B  A9 01             SEND: LDA   #SUCCES     ;ASSUME SUCCESS
1960  EA6D  85 30                   STA   STATUS
1961                                ;
1962  EA6F  20 F6 EB                JSR   SENDEN      ;ENABLE SENDING
1963                                ;
1964  EA72  A0 00                   LDY   #0
1965  EA74  84 31                   STY   CHKSUM      ;CLEAR CHECK SUM
1966  EA76  84 3B                   STY   CHKSNT      ;CHECKSUM SENT FLAG
1967  EA78  84 3A                   STY   XMTDON      ;TRANSMISSION DONE FLAG
1968                                ;
1969                                ;
```

```
1970  EA7A  B1 32          LDA   (BUFRLO),Y  ;PUT FIRST BYTE FROM BUFFER
1971  EA7C  8D 0D D2       STA   SEROUT      ;INTO THE SERIAL OUTPUT REGISTER
1972                       ;
1973                       ;
1974  EA7F  85 31          STA   CHKSUM      ;PUT IT IN CHECKSUM
1975                       ;
1976  EA81  A5 11          NOTDON: LDA   BRKKEY
1977  EA83  D0 03          BNE   NTBRKO
1978  EA85  4C A4 ED       JMP   BROKE       ;JUMP IF BREAK KEY PRESSED
1979                       ;
1980  EA88  A5 3A          NTBRKO: LDA   XMTDON      ;LOOP UNTIL TRANSMISSION IS DONE
1981  EA8A  F0 F5          BEQ   NOTDON
1982                       ;
1983  EA8C  20 63 EC       JSR   SENDDS      ;DISABLE SENDING
1984                       ;
1985  EA8F  60             RTS               ;RETURN
1986                       ;
1987                       ;
1988                       ;
1989                       ;
1990                       ;
1991                       ;
1992                       ;OUTPUT DATA NEEDED INTERRUPT SERVICE ROUTINE
1993                       ;
1994  EA90  98             ISRODN: TYA
1995  EA91  48             PHA               ;SAVE Y REG ON STACK
1996                       ;
1997  EA92  E6 32          INC   BUFRLO      ;INCREMENT BUFFER POINTER
1998  EA94  D0 02          BNE   NOWRPO
1999  EA96  E6 33          INC   BUFRHI
2000                       ;
2001  EA98  A5 33          NOWRPO: LDA   BUFRHI      ;CHECK IF PAST END OF BUFFER
2002  EA9A  C5 35          CMP   BFENHI
2003  EA9C  90 22          BCC   NOTEND
2004  EA9E  A5 32          LDA   BUFRLO
2005  EAA0  C5 34          CMP   BFENLO
2006  EAA2  90 1C          BCC   NOTEND      ;BRANCH IF NOT PAST END OF BUFFER
2007                       ;
2008  EAA4  A5 3B          LDA   CHKSNT
2009  EAA6  D0 0B          BNE   RELONE      ;BRANCH IF CHECKSUM ALREADY SENT
2010                       ;
2011  EAA8  A5 31          LDA   CHKSUM
2012  EAAA  8D 0D D2       STA   SEROUT      ;SEND CHECK SUM
2013  EAAD  A9 FF          LDA   #$FF
2014  EAAF  85 3B          STA   CHKSNT      ;SET CHECKSUM SENT FLAG
2015  EAB1  D0 09          BNE   CHKDON
2016                       ;
2017  EAB3  A5 10          RELONE: LDA   POKMSK      ;ENABLE TRANSMIT DONE INTERRUPT
2018  EAB5  09 08          ORA   #$08
2019  EAB7  85 10          STA   POKMSK
2020  EAB9  8D 0E D2       STA   IRQEN
2021                       ;
2022  EABC  68             CHKDON: PLA
2023  EABD  A8             TAY               ;RESTORE Y REG
```

```
2024  EABE  68          PLA             ;RETURN FROM INTERRUPT
2025  EABF  40          RTI
2026                    ;
2027                    ;
2028  EAC0  A0 00       NOTEND: LDY     #0
2029  EAC2  B1 32               LDA     (BUFRLO),Y  ;PUT NEXT BYTE FROM BUFFER
2030  EAC4  8D 0D D2            STA     SEROUT      ;INTO THE SERIAL OUTPUT REGISTER
2031                    ;
2032  EAC7  18                  CLC                 ;ADD IT TO CHECKSUM
2033  EAC8  65 31               ADC     CHKSUM
2034  EACA  69 00               ADC     #0
2035  EACC  85 31               STA     CHKSUM
2036                    ;
2037  EACE  4C BC EA            JMP     CHKDON      ;GO RETURN
2038                    ;
2039                    ;
2040                    ;
2041                    ;
2042                    ;
2043                    ;
2044                    ;TRANSMIT DONE INTERRUPT SERVICE ROUTINE
2045                    ;
2046  EAD1  A5 3B       ISRTD:  LDA     CHKSNT
2047  EAD3  F0 0B               BEQ     FOOEY       ;BRANCH IF CHECKSUM NOT YET SENT
2048                    ;
2049  EAD5  85 3A               STA     XMTDON      ;OTHERWISE SET TRANSMISSION DONE FLAG
2050                    ;
2051  EAD7  A5 10               LDA     POKMSK      ;DISABLE TRANSMIT DONE INTERRUPT
2052  EAD9  29 F7               AND     #$F7
2053  EADB  85 10               STA     POKMSK
2054  EADD  8D 0E D2            STA     IRQEN
2055                    ;
2056  EAE0  68          FOOEY:  PLA             ;RETURN FROM INTERRUPT
2057  EAE1  40                  RTI
2058                    ;
2059                    ;
2060                    ;
2061                    ;
2062                    ;
2063                    ;
2064                    ;
2065                    ;
2066                    ;RECEIVE SUBROUTINE
2067                    ;
2068  EAE2  A9 00       RECEIV: LDA     #0
2069                    ;
2070  EAE4  AC 0F 03            LDY     CASFLG
2071  EAE7  D0 02               BNE     NOCLR       ;BRANCH IF CASSETTE
2072                    ;
2073  EAE9  85 31               STA     CHKSUM      ;CLEAR CHKSUM
2074  EAEB  85 38       NOCLR:  STA     BUFRFL      ;BUFFER FULL FLAG
2075  EAED  85 39               STA     RECVDN      ;RECEIVE DONE FLAG
2076                    ;
2077                    ;
```

```
2078                            ;
2079  EAEF  A9 01               LDA     #SUCCES
2080  EAF1  85 30               STA     STATUS      ;SET GOOD STATUS FOR DEFAULT CASE.
2081  EAF3  20 1F EC            JSR     RECVEN      ;DO RECEIVE ENABLE
2082  EAF6  A9 3C               LDA     #NCOMHI     ;COMMAND FRAME HI COMMAND
2083  EAF8  8D 03 D3            STA     PBCTL       ;STORE IN PIA
2084  EAFB  A5 11       CHKTIM: LDA     BRKKEY
2085  EAFD  D0 03               BNE     NTBRK1
2086  EAFF  4C A4 ED            JMP     BROKE       ;JUMP IF BREAK KEY PRESSED
2087                            ;
2088  EB02  AD 17 03    NTBRK1: LDA     TIMFLG      ;NO,
2089  EB05  F0 05               BEQ     TOUT        ;IF TIMEOUT, GO SET ERROR STATUS
2090  EB07  A5 39               LDA     RECVDN
2091  EB09  F0 F0               BEQ     CHKTIM      ;DONE ?
2092  EB0B  60          GOBACK: RTS
2093  EB0C  A9 8A       TOUT:   LDA     #TIMOUT     ;YES,
2094  EB0E  85 30               STA     STATUS      ;SET TIMEOUT STATUS
2095                            ;
2096                            ;
2097                            ;
2098                            ;
2099                            ;
2100                            ;
2101  EB10  60          RRETRN: RTS                 ;RETURN
2102                            ;
2103                            ;
2104                            ;
2105                            ;
2106                            ;
2107                            ;
2108                            ;
2109                            ;SERIAL INPUT READY INTERRUPT SERVICE ROUTINE
2110                            ;
2111  EB11  98          ISRSIR: TYA
2112  EB12  48                  PHA                 ;SAVE Y REG ON STACK
2113                            ;
2114                            ;
2115                            ;
2116  EB13  AD 0F D2            LDA     SKSTAT
2117  EB16  8D 0A D2            STA     SKRES       ;RESET STATUS REGISTER
2118                            ;********   THIS MAY NOT BE THE PLACE TO DO IT   *********
2119                            ;
2120  EB19  30 04               BMI     NTFRAM      ;BRANCH IF NO FRAMING ERROR
2121                            ;
2122  EB1B  A0 8C               LDY     #FRMERR
2123  EB1D  84 30               STY     STATUS      ;SET FRAME ERRORR STATUS
2124                            ;
2125  EB1F  29 20       NTFRAM: AND     #$20
2126  EB21  D0 04               BNE     NTOVRN      ;BRANCH IF NO OVERRUN ERROR
2127                            ;
2128  EB23  A0 8E               LDY     #OVRRUN
2129  EB25  84 30               STY     STATUS      ;SET OVERRUN ERROR STATUS
2130                            ;
2131  EB27  A5 38       NTOVRN: LDA     BUFRFL
```

```
2132 EB29 F0 13          BEQ     NOTYET      ;BRANCH IF BUFFER WAS NOT YET FILLED
2133                     ;
2134 EB2B AD 0D D2       LDA     SERIN       ;THIS INPUT BYTE IS THE CHECKSUM
2135 EB2E C5 31          CMP     CHKSUM
2136 EB30 F0 04          BEQ     SRETRN      ;BRANCH IF CHECKSUMS MATCH
2137                     ;
2138 EB32 A0 8F          LDY     #CHKERR
2139 EB34 84 30          STY     STATUS      ;SET CHECKSUM ERROR STATUS
2140                     ;
2141 EB36 A9 FF          SRETRN: LDA     #$FF        ;SET RECEIVE DONE FLAG
2142 EB38 85 39          STA     RECVDN
2143                     ;
2144 EB3A 68             SUSUAL: PLA
2145 EB3B A8                     TAY             ;RESTORE Y REG
2146 EB3C 68                     PLA             ;RETURN FROM INTERRUPT
2147 EB3D 40                     RTI
2148                     ;
2149                     ;
2150                     ;
2151 EB3E AD 0D D2       NOTYET: LDA     SERIN
2152 EB41 A0 00          LDY     #0
2153 EB43 91 32          STA     (BUFRLO),Y  ;STORE INPUT REGISTER INTO BUFFER
2154                     ;
2155 EB45 18             CLC             ;ADD IT TO CHECKSUM
2156 EB46 65 31          ADC     CHKSUM
2157 EB48 69 00          ADC     #0
2158 EB4A 85 31          STA     CHKSUM
2159                     ;
2160 EB4C E6 32          INC     BUFRLO      ;INCREMENT BUFFER POINTER
2161 EB4E D0 02          BNE     NTWRP1
2162 EB50 E6 33          INC     BUFRHI
2163                     ;
2164 EB52 A5 33          NTWRP1: LDA     BUFRHI
2165 EB54 C5 35          CMP     BFENHI
2166 EB56 90 E2          BCC     SUSUAL
2167 EB58 A5 32          LDA     BUFRLO
2168 EB5A C5 34          CMP     BFENLO
2169 EB5C 90 DC          BCC     SUSUAL      ;BRANCH IF NEW BUFFER ADDRESS IS IN BUFFER L
2170                     ;
2171 EB5E A5 3C          LDA     NOCKSM
2172 EB60 F0 06          BEQ     GOON        ;BRANCH IF A CHECKSUM WILL FOLLOW DATA
2173                     ;
2174 EB62 A9 00          LDA     #0
2175 EB64 85 3C          STA     NOCKSM      ;CLEAR NO CHECKSUM FLAG
2176                     ;
2177 EB66 F0 CE          BEQ     SRETRN      ;GO RETURN AND SET RECEIVE DONE FLAG
2178                     ;
2179                     ;
2180 EB68 A9 FF          GOON:   LDA     #$FF
2181 EB6A 85 38          STA     BUFRFL      ;SET BUFFER FULL FLAG
2182                     ;
2183 EB6C D0 CC          BNE     SUSUAL      ;GO RETURN
2184                     ;
2185                     ;
```

```
2186                                ;
2187                                ;
2188                                ;
2189                                ;
2190                                ;
2191                                ;
2192                                ;LOAD BUFFER POINTER SUBROUTINE
2193                                ;
2194                                ;LOAD BUFFER POINTER WITH DCB BUFFER INFORMATION
2195                                ;
2196 EB6E 18                       LDPNTR: CLC
2197 EB6F AD 04 03                         LDA     DBUFLO
2198 EB72 85 32                            STA     BUFRLO
2199 EB74 6D 08 03                         ADC     DBYTLO
2200 EB77 85 34                            STA     BFENLO      ;ALSO SET BUFFER END + 1 ADDRESS
2201                                ;
2202 EB79 AD 05 03                         LDA     DBUFHI
2203 EB7C 85 33                            STA     BUFRHI
2204 EB7E 6D 09 03                         ADC     DBYTHI
2205 EB81 85 35                            STA     BFENHI
2206                                ;
2207 EB83 60                               RTS                 ;RETURN
2208                                ;
2209                                ;
2210                                ;
2211                                ;
2212                                ;
2213                                ;
2214                                ;
2215                                ;
2216                                ;CASSETTE HANDLING CODE
2217                                ;
2218 EB84 AD 03 03                 CASENT: LDA     DSTATS
2219 EB87 10 2E                            BPL     CASRED      ;BRANCH IF INPUT FROM CASSETTE
2220                                ;
2221                                ;WRITE A RECORD
2222                                ;
2223 EB89 A9 CC                            LDA     #B600LO     ;SET BAUD RATE TO 600
2224 EB8B 8D 04 D2                         STA     AUDF3
2225 EB8E A9 05                            LDA     #B600HI
2226 EB90 8D 06 D2                         STA     AUDF4
2227                                ;
2228 EB93 20 F6 EB                         JSR     SENDEN      ;TURN ON POKEY MARK TONE
2229                                ;
2230 EB96 A0 0F                            LDY     #WSIRG      ;LOAD SHORT WRITE INTER RECORD GAP TIME
2231 EB98 AD 0B 03                         LDA     DAUX2
2232 EB9B 30 02                            BMI     SRTIRO      ;BRANCH IF SHORT GAP IS DESIRED
2233                                ;
2234 EB9D A0 B4                            LDY     #WIRGLO     ;SET WRITE IRG TIME
2235 EB9F A2 00                    SRTIRO: LDX     #WIRGHI
2236 EBA1 20 BD ED                         JSR     SETVBX
2237                                ;
2238 EBA4 A9 34                            LDA     #MOTRGO
2239 EBA6 8D 02 D3                         STA     PACTL       ;TURN ON MOTOR
```

```
2240                            ;
2241  EBA9  AD 17 03    TIMIT:  LDA     TIMFLG      ;LOOP UNTIL DONE
2242  EBAC  D0 FB               BNE     TIMIT
2243                            ;
2244  EBAE  20 6E EB            JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
2245                            ;
2246  EBB1  20 6B EA            JSR     SEND        ;SEND A BUFFER
2247                            ;
2248  EBB4  4C E3 EB            JMP     CRETRN      ;GO RETURN
2249                            ;
2250                            ;
2251                            ;
2252                            ;RECEIVE A RECORD
2253                            ;
2254  EBB7  A9 FF       CASRED: LDA     #$FF
2255  EBB9  8D 0F 03            STA     CASFLG      ;SET SET CASSETTE FLAG
2256                            ;
2257  EBBC  A0 0A               LDY     #RSIRG      ;LOAD SHORT READ INTER RECORD GAP TIME
2258  EBBE  AD 0B 03            LDA     DAUX2
2259  EBC1  30 02               BMI     SRTIR1      ;BRANCH IF SHORT GAP IS DESIRED
2260                            ;
2261  EBC3  A0 78               LDY     #RIRGLO     ;SET TIME OUT FOR READ IRG
2262  EBC5  A2 00       SRTIR1: LDX     #RIRGHI
2263  EBC7  20 BD ED            JSR     SETVBX
2264                            ;
2265  EBCA  A9 34               LDA     #MOTRGO
2266  EBCC  8D 02 D3            STA     PACTL       ;TURN ON MOTOR
2267                            ;
2268  EBCF  AD 17 03    TIMIT1: LDA     TIMFLG      ;LOOP UNTIL DONE
2269  EBD2  D0 FB               BNE     TIMIT1
2270                            ;
2271  EBD4  20 6E EB            JSR     LDPNTR      ;LOAD BUFFER POINTER WITH DCB INFORMATION
2272                            ;
2273  EBD7  20 79 EC            JSR     STTMOT      ;SET DEVICE TIME OUT IN Y,X
2274  EBDA  20 BD ED            JSR     SETVBX
2275                            ;
2276  EBDD  20 14 ED            JSR     BEGIN       ;SET INITIAL BAUD RATE
2277                            ;
2278  EBE0  20 E2 EA            JSR     RECEIV      ;GO RECEIVE A BLOCK
2279                            ;
2280  EBE3  AD 0B 03    CRETRN: LDA     DAUX2
2281  EBE6  30 05               BMI     SRTIR2      ;BRANCH IF DOING SHORT INTER RECORD GAPS
2282                            ; DON'T TURN OFF CASSETTE MOTOR
2283  EBE8  A9 3C               LDA     #MOTRST
2284  EBEA  8D 02 D3            STA     PACTL       ;TURN OFF MOTOR
2285                            ;
2286  EBED  4C 0D EA    SRTIR2: JMP     RETURN      ;GO RETURN
2287                            ;
2288                            ;
2289                            ;
2290                            ;
2291                            ;
2292  EBF0  A9 00       JTIMER: LDA     #$00
2293  00EB              JTADRH  =       JTIMER/256  ;HI BYTE OF JUMP TIMER ROUTINE ADDR
```

```
2294  00F0                    JTADRL  =       (-256)*JTADRH+JTIMER
2295  EBF2  8D 17 03                  STA     TIMFLG      ;SET TIME OUT FLAG
2296  EBF5  60                        RTS
2297                          ;
2298                          ;
2299                          ;
2300                          ;
2301                          ;
2302                          ;
2303                          ;SEND ENABLE SUBROUTINE
2304                          ;
2305  EBF6  A9 07             SENDEN: LDA     #$07        ;MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
2306  EBF8  2D 32 02                  AND     SSKCTL
2307  EBFB  09 20                     ORA     #$20        ;SET TRANSMIT MODE
2308                          ;
2309  EBFD  AC 00 03                  LDY     DDEVIC
2310  EC00  C0 60                     CPY     #CASET
2311  EC02  D0 0C                     BNE     NOTCAS      ;BRANCH IF NOT CASSETTE
2312                          ;
2313  EC04  09 08                     ORA     #$08        ;SET THE FSK OUTPUT BIT
2314                          ;
2315  EC06  A0 07                     LDY     #LOTONE     ;SET FSK TONE FREQUENCIES
2316  EC08  8C 02 D2                  STY     AUDF2
2317  EC0B  A0 05                     LDY     #HITONE
2318  EC0D  8C 00 D2                  STY     AUDF1
2319                          ;
2320  EC10  8D 32 02          NOTCAS: STA     SSKCTL      ;STORE NEW VALUE TO SYSTEM MASK
2321  EC13  8D 0F D2                  STA     SKCTL       ;STORE TO ACTUAL REGISTER
2322                          ;
2323  EC16  A9 C7                     LDA     #$C7        ;MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
2324  EC18  25 10                     AND     POKMSK
2325  EC1A  09 10                     ORA     #$10        ;ENABLE OUTPUT DATA NEEDED INTERRUPT
2326                          ;
2327                          ;
2328  EC1C  4C 35 EC                  JMP     CONTIN      ;GO CONTINUE IN RECEIVE ENABLE SUBROUTINE
2329                          ;
2330                          ;
2331                          ;
2332                          ;
2333                          ;
2334                          ;
2335                          ;
2336                          ;
2337                          ;
2338                          ;
2339                          ;RECEIVE ENABLE SUBROUTINE
2340                          ;
2341  EC1F  A9 07             RECVEN: LDA     #$07        ;MASK OFF PREVIOUS SERIAL BUS CONTROL BITS
2342  EC21  2D 32 02                  AND     SSKCTL
2343  EC24  09 10                     ORA     #$10        ;SET RECEIVE MODE ASYNCH.
2344  EC26  8D 32 02                  STA     SSKCTL      ;STORE NEW VALUE TO SYSTEM MASK
2345  EC29  8D 0F D2                  STA     SKCTL       ;STORE TO ACTUAL REGISTER
2346
2347  EC2C  8D 0A D2                  STA     SKRES       ;RESET SERIAL PORT/KEYBOARD STATUS REGISTER
```

```
2348                            ;
2349  EC2F  A9 C7               LDA     #$C7        ;MASK OFF PREVIOUS SERIAL BUS INTERRUPT BITS
2350  EC31  25 10               AND     POKMSK
2351  EC33  09 20               ORA     #$20        ;ENABLE RECEIVE INTERRUPT
2352  EC35  85 10     CONTIN:   STA     POKMSK      ;STORE NEW VALUE TO SYSTEM MASK
2353  EC37  8D 0E D2            STA     IRQEN       ;STORE TO ACTUAL REGISTER
2354                            ;
2355                            ;
2356  EC3A  A9 28               LDA     #$28        ;CLOCK CH.3 WITH 1.79 MHZ
2357  EC3C  8D 08 D2            STA     AUDCTL      ;CLOCK CH.4 WITH CH. 3
2358                            ;
2359  EC3F  A2 06               LDX     #6          ;SET PURE TONES, NO VOLUME
2360  EC41  A9 A8               LDA     #$A8
2361  EC43  A4 41               LDY     SOUNDR      ;TEST QUIET I/O FLAG
2362  EC45  D0 02               BNE     NOISE1      ;NE IS NORMAL (NOISY)
2363  EC47  A9 A0               LDA     #$A0
2364  EC49  9D 01 D2  NOISE1:  STA     AUDC1,X
2365  EC4C  CA                  DEX
2366  EC4D  CA                  DEX
2367  EC4E  10 F9               BPL     NOISE1
2368                            ;
2369  EC50  A9 A0               LDA     #$A0
2370  EC52  8D 05 D2            STA     AUDC3       ;TURN OFF SOUND ON CHANNEL 3
2371  EC55  AC 00 03            LDY     DDEVIC
2372  EC58  C0 60               CPY     #CASET
2373  EC5A  F0 06               BEQ     CAS31       ;BRANCH IF CASSETTE IS DESIRED
2374  EC5C  8D 01 D2            STA     AUDC1       ;OTHERWISE TURN OFF CHANNELS 1 AND 2
2375  EC5F  8D 03 D2            STA     AUDC2
2376                            ;
2377                            ;
2378  EC62  60        CAS31:    RTS                 ;RETURN
2379                            ;
2380                            ;
2381                            ;
2382                            ;
2383                            ;
2384                            ;
2385                            ;
2386                            ;
2387                            ;
2388                            ;
2389                            ;DISABLE SEND AND DISABLE RECEIVE SUBROUTINES
2390                            ;
2391  EC63  EA        SENDDS:   NOP
2392  EC64  A9 C7     RECVDS:   LDA     #$C7        ;MASK OFF SERIAL BUS INTERRUPTS
2393  EC66  25 10               AND     POKMSK
2394  EC68  85 10               STA     POKMSK      ;STORE NEW VALUE TO SYSTEM MASK
2395  EC6A  8D 0E D2            STA     IRQEN       ;STORE TO ACTUAL REGISTER
2396                            ;
2397  EC6D  A2 06               LDX     #6
2398  EC6F  A9 00               LDA     #0
2399  EC71  9D 01 D2  ZERIT:    STA     AUDC1,X
2400  EC74  CA                  DEX
2401  EC75  CA                  DEX
```

```
2402  EC76  10 F9              BPL     ZERIT       ;TURN OFF AUDIO VOLUME
2403                          ;
2404  EC78  60                 RTS                 ;RETURN
2405                          ;
2406                          ;
2407                          ;
2408                          ;
2409                          ;
2410                          ;
2411                          ;
2412                          ;
2413                          ;
2414                          ;
2415                          ;SET DDEVICE TIME OUT VALUES IN Y,X SUBROUTINE
2416                          ;
2417  EC79  AD 06 03           STTMOT: LDA     DTIMLO      ;GET DEVICE TIME OUT IN 1 SECOND INCR
2418  EC7C  6A                 ROR     A           ;PUT 6 HI BITS IN X, LO 2 BITS IN Y
2419  EC7D  6A                 ROR     A
2420  EC7E  A8                 TAY                 ;TEMP SAVE
2421  EC7F  29 3F              AND     #$3F        ;MASK OFF 2 HI BITS
2422  EC81  AA                 TAX                 ;THIS IS HI BYTE OF TIME OUT
2423                          ;
2424  EC82  98                 TYA                 ;RESTORE
2425  EC83  6A                 ROR     A
2426  EC84  29 C0              AND     #$C0        ;MASK OFF ALL BUT 2 HI BITS
2427  EC86  A8                 TAY                 ;THIS IS LO BYTE OF TIME OUT
2428                          ;
2429  EC87  60                 RTS
2430                          ;
2431                          ;
2432                          ;
2433                          ;
2434                          ;
2435                          ;
2436                          ;
2437                          ;
2438                          ;
2439                          ;
2440  EC88  11 EB              INTTBL: .WORD   ISRSIR      ;SERIAL INPUT READY
2441  EC8A  90 EA                      .WORD   ISRODN      ;OUTPUT DATA NEEDED
2442  EC8C  D1 EA                      .WORD   ISRTD       ;TRANSMISSION DONE
2443                          ;
2444  00EB                    SIRHI   =       ISRSIR/256  ;SERIAL INPUT READY ISR ADDRESS
2445  0011                    SIRLO   =       (-256)*SIRHI+ISRSIR
2446  00EA                    ODNHI   =       ISRODN/256  ;OUTPUT DATA NEEDED ISR ADDRESS
2447  0090                    ODNLO   =       (-256)*ODNHI+ISRODN
2448  00EA                    TDHI    =       ISRTD/256   ;TRANSMISSION DONE ISR ADDRESS
2449  00D1                    TDLO    =       (-256)*TDHI+ISRTD
2450                          ;
2451                          ;
2452                          ;
2453                          ;
2454                          ;SEND A DATA FRAME TO AN INTELLIGENT PERIPHERAL SUBROUTINE
2455                          ;
```

```
2456                            ;
2457  EC8E  A2 01               SENDIN: LDX     #$01
2458  EC90  A0 FF               DELAY0: LDY     #$FF
2459  EC92  88                  DELAY1: DEY
2460  EC93  D0 FD                       BNE     DELAY1
2461  EC95  CA                          DEX
2462  EC96  D0 F8                       BNE     DELAY0
2463                            ;
2464  EC98  20 6B EA                    JSR     SEND            ;GO SEND THE DATA FRAME
2465                            ;
2466  EC9B  A0 02                       LDY     #CTIMLO         ;SET ACK TIME OUT
2467  EC9D  A2 00                       LDX     #CTIMHI
2468  EC9F  20 BD ED            WAITER: JSR     SETVBX
2469                            ;
2470  ECA2  20 1A EA                    JSR     WAIT            ;WAIT FOR ACK
2471                            ;
2472  ECA5  98                          TYA                     ;IF Y=0, A TIME OUT OR NACK OCCURED
2473                            ;
2474  ECA6  60                          RTS                     ;RETURN
2475                            ;
2476                            ;
2477                            ;
2478                            ;
2479                            ;
2480                            ;
2481                            ;
2482                            ;
2483                            ;
2484                            ;
2485                            ;
2486                            ; COMPUTE VALUE FOR POKEY FREQ REGS FOR THE BAUD RATE AS
2487                            ; MEASURED BY AN INTERVAL OF THE 'VCOUNT' TIMER.
2488                            ;
2489  ECA7  8D 10 03            COMPUT: STA     TIMER2
2490  ECAA  8C 11 03                    STY     TIMER2+1        ;SAVE FINAL TIMER VALUE
2491  ECAD  20 08 ED                    JSR     ADJUST          ;ADJUST VCOUNT VALUE
2492  ECB0  8D 10 03                    STA     TIMER2          ;SAVE ADJUSTED VALUE
2493  ECB3  AD 0C 03                    LDA     TIMER1
2494  ECB6  20 08 ED                    JSR     ADJUST          ;ADJUST
2495  ECB9  8D 0C 03                    STA     TIMER1          ;SAVE ADJUSTED TIMER1 VALUE
2496  ECBC  AD 10 03                    LDA     TIMER2
2497  ECBF  38                          SEC
2498  ECC0  ED 0C 03                    SBC     TIMER1
2499  ECC3  8D 12 03                    STA     TEMP1           ;FIND VCOUNT DIFFERENCE
2500  ECC6  AD 11 03                    LDA     TIMER2+1
2501  ECC9  38                          SEC
2502  ECCA  ED 0D 03                    SBC     TIMER1+1
2503  ECCD  A8                          TAY                     ;FIND VBLANK COUNT DIFFERENCE
2504  ECCE  A9 7D                       LDA     #-$83
2505  ECD0  18                  HITIMR: CLC
2506  ECD1  69 83                       ADC     #$83            ;ACCUMULATE MULTIPLICATION
2507  ECD3  88                          DEY
2508  ECD4  10 FA                       BPL     HITIMR          ;DONE?
2509  ECD6  18                          CLC
```

```
2510 ECD7 6D 12 03          ADC   TEMP1       ;TOTAL VCOUNT DIFFERENCE
2511 ECDA A8          FINDX: TAY              ;SAVE ACCUM
2512 ECDB 4A                 LSR   A
2513 ECDC 4A                 LSR   A
2514 ECDD 4A                 LSR   A
2515 ECDE 0A                 ASL   A
2516 ECDF 38                 SEC
2517 ECE0 E9 16              SBC   #22         ;ADJUST TABLE INDEX
2518 ECE2 AA                 TAX               ;DIVIDE INTERVAL BY 4 TO GET TABLE INDEX
2519 ECE3 98                 TYA               ;RESTORE ACCUM
2520 ECE4 29 07              AND   #7
2521 ECE6 A8                 TAY               ;PULL OFF 3 LO BITS OF INTERVAL
2522 ECE7 A9 F5              LDA   #-11
2523 ECE9 18          DOINTP: CLC
2524 ECEA 69 0B              ADC   #11         ;ACCUMULATE INTERPOLATION CONSTANT
2525 ECEC 88                 DEY
2526 ECED 10 FA              BPL   DOINTP      ;INTERPOLATION CONSTANT COMPUTATION DONE?
2527                    ;
2528 ECEF A0 00       ENINTP: LDY   #0
2529 ECF1 8C 0E 03           STY   ADDCOR      ;CLEAR ADDITION CORRECTION FLAG
2530 ECF4 38                 SEC
2531 ECF5 E9 07              SBC   #7          ;ADJUST INTERPOLATION CONSTANT
2532 ECF7 10 03              BPL   PLUS
2533 ECF9 CE 0E 03           DEC   ADDCOR
2534 ECFC 18          PLUS:   CLC
2535 ECFD 7D D2 ED           ADC   POKTAB,X    ;ADD CONSTANT TO LO BYTE TABLE VALUE
2536 ED00 A8                 TAY               ;LO BYTE POKEY FREQ VALUE
2537 ED01 AD 0E 03           LDA   ADDCOR
2538 ED04 7D D3 ED           ADC   POKTAB+1,X  ;ADD CARRY TO HI BYTE TABLE VALUE
2539                    ; HI BYTE POKEY FREQ VALUE
2540 ED07 60                 RTS
2541                    ;
2542                    ;
2543                    ;
2544                    ;       ROUTINE TO ADJUST VCOUNT VALUE
2545                    ;
2546 ED08 C9 7C       ADJUST: CMP   #$7C
2547 ED0A 30 04              BMI   ADJ1        ;LARGER THAN '7C' ?
2548 ED0C 38                 SEC               ;YES,
2549 ED0D E9 7C              SBC   #$7C
2550 ED0F 60                 RTS
2551 ED10 18          ADJ1:   CLC
2552 ED11 69 07              ADC   #$7
2553 ED13 60                 RTS
2554                    ;
2555                    ;
2556                    ;
2557                    ;
2558                    ;
2559                    ;
2560                    ;
2561                    ;       INITIAL BAUD RATE MEASUREMENT -- USED TO SET THE
2562                    ;               BAUD RATE AT THE START OF A RECORD.
2563                    ;
```

```
2564                            ;               IT IS ASSUMED THAT THE FIRST TWO BYTES OF EVERY
2565                            ;               RECORD ARE 'AA' HEX.
2566                            ;
2567  ED14  A5 11               BEGIN:  LDA     BRKKEY
2568  ED16  D0 03                       BNE     NTBRK2
2569  ED18  4C A4 ED                    JMP     BROKE           ;JUMP IF BREAK KEY PRESSED
2570                            ;
2571  ED1B  78                  NTBRK2: SEI
2572                            ;
2573  ED1C  AD 17 03                    LDA     TIMFLG
2574  ED1F  D0 02                       BNE     OKTIM1          ;BRANCH IF NOT TIMED OUT
2575  ED21  F0 25                       BEQ     TOUT1           ;BRANCH IF TIME OUT
2576                            ;
2577  ED23  AD 0F D2            OKTIM1: LDA     SKSTAT
2578  ED26  29 10                       AND     #$10            ;READ SERIAL PORT
2579  ED28  D0 EA                       BNE     BEGIN           ;START BIT?
2580  ED2A  8D 16 03                    STA     SAVIO           ;SAVE SER. DATA IN
2581  ED2D  AE 0B D4                    LDX     VCOUNT          ;READ VERTICAL LINE COUNTER
2582  ED30  A4 14                       LDY     RTCLOK+2        ;READ LO BYTE OF VBLANK CLOCK
2583  ED32  8E 0C 03                    STX     TIMER1
2584  ED35  8C 0D 03                    STY     TIMER1+1        ;SAVE INITIAL TIMER VALUE
2585                            ;
2586  ED38  A2 01                       LDX     #1              ;SET MODE FLAG
2587  ED3A  8E 15 03                    STX     TEMP3
2588  ED3D  A0 0A                       LDY     #10             ;SET BIT COUNTER FOR 10 BITS
2589  ED3F  A5 11               COUNT:  LDA     BRKKEY
2590  ED41  F0 61                       BEQ     BROKE           ;BRANCH IF BREAK KEY PRESSED
2591                            ;
2592  ED43  AD 17 03                    LDA     TIMFLG
2593  ED46  D0 04                       BNE     OKTIMR          ;BRANCH IF NOT TIMED OUT
2594  ED48  58                  TOUT1:  CLI
2595  ED49  4C 0C EB                    JMP     TOUT            ;BRANCH IF TIME OUT
2596                            ;
2597  ED4C  AD 0F D2            OKTIMR: LDA     SKSTAT
2598  ED4F  29 10                       AND     #$10            ;READ SERIAL PORT
2599  ED51  CD 16 03                    CMP     SAVIO           ;DATA IN CHANGED YET?
2600  ED54  F0 E9                       BEQ     COUNT
2601  ED56  8D 16 03                    STA     SAVIO           ;YES, SAVE SER. DATA IN
2602  ED59  88                          DEY                     ;DECR. BIT COUNTER
2603  ED5A  D0 E3                       BNE     COUNT           ;DONE?
2604                            ;
2605  ED5C  CE 15 03                    DEC     TEMP3           ;YES,
2606  ED5F  30 12                       BMI     GOREAD          ;DONE WITH BOTH MODES?
2607  ED61  AD 0B D4                    LDA     VCOUNT
2608  ED64  A4 14                       LDY     RTCLOK+2        ;READ TIMER LO & HI BYTES
2609  ED66  20 A7 EC                    JSR     COMPUT          ;NO, COMPUTE BAUD RATE
2610  ED69  8C EE 02                    STY     CBAUDL
2611  ED6C  8D EF 02                    STA     CBAUDH          ;SET BAUD RATE INTO RAM CELLS
2612  ED6F  A0 09                       LDY     #9              ;SET BIT COUNTER FOR 9 BITS
2613  ED71  D0 CC                       BNE     COUNT
2614                            ;
2615  ED73  AD EE 02            GOREAD: LDA     CBAUDL
2616  ED76  8D 04 D2                    STA     AUDF3
2617  ED79  AD EF 02                    LDA     CBAUDH
```

```
2618 ED7C 8D 06 D2          STA   AUDF4        ;SET POKEY FREQ REGS FOR BAUD RATE
2619 ED7F A9 00             LDA   #0
2620 ED81 8D 0F D2          STA   SKSTAT
2621 ED84 AD 32 02          LDA   SSKCTL
2622 ED87 8D 0F D2          STA   SKSTAT       ;INIT. POKEY SERIAL PORT
2623 ED8A A9 55             LDA   #$55
2624 ED8C 91 32             STA   (BUFRLO),Y   ;STORE '$55' AS FIRST RCV. BUFFER
2625 ED8E C8                INY
2626 ED8F 91 32             STA   (BUFRLO),Y
2627 ED91 A9 AA             LDA   #$AA
2628 ED93 85 31             STA   CHKSUM       ;STORE CHECKSUM FOR 2 BYTES OF '$AA'
2629 ED95 18                CLC
2630 ED96 A5 32             LDA   BUFRLO
2631 ED98 69 02             ADC   #2
2632 ED9A 85 32             STA   BUFRLO
2633 ED9C A5 33             LDA   BUFRHI
2634 ED9E 69 00             ADC   #0
2635 EDA0 85 33             STA   BUFRHI       ;INCR. BUFFER POINTER BY 1
2636 EDA2 58                CLI
2637 EDA3 60                RTS
2638                    ;
2639                    ;
2640                    ;
2641 EDA4 20 63 EC      BROKE: JSR   SENDDS     ;BREAK KEY WAS PRESSED, SO PREPARE
2642 EDA7 A9 3C             LDA   #MOTRST      ;TO RETURN
2643 EDA9 8D 02 D3          STA   PACTL        ;TURN OFF MOTOR
2644 EDAC 8D 03 D3          STA   PBCTL        ;RAISE NOT COMMAND LINE
2645                    ;
2646 EDAF A9 80             LDA   #BRKABT
2647 EDB1 85 30             STA   STATUS       ;STORE BREAK ABORT STATUS CODE
2648                    ;
2649 EDB3 AE 18 03          LDX   STACKP
2650 EDB6 9A                TXS                ;RESTORE STACK POINTER
2651                    ;
2652 EDB7 C6 11             DEC   BRKKEY       ;SET BREAK KEY FLAG TO NONZERO
2653 EDB9 58                CLI                ;ALLOW IRG'S
2654                    ;
2655 EDBA 4C 0D EA          JMP   RETURN       ;GO RETURN
2656                    ;
2657                    ;
2658                    ;
2659                    ;
2660                    ;
2661 EDBD A9 F0        SETVBX: LDA   #JTADRL    ;STORE TIME OUT ROUTINE ADDRESS
2662 EDBF 8D 26 02          STA   CDTMA1
2663 EDC2 A9 EB             LDA   #JTADRH
2664 EDC4 8D 27 02          STA   CDTMA1+1
2665                    ;
2666 EDC7 A9 01             LDA   #1           ;SET FOR TIMER 1
2667 EDC9 8D 17 03          STA   TIMFLG       ;SET FLAG TO NOT TIMED OUT
2668                    ;
2669 EDCC 78                SEI                ;THE SETVBL ROUTINE NEEDS THIS TO CUT SHORT
2670 EDCD 20 5C E4          JSR   SETVBV       ;ANY VBLANKS THAT OCCUR
2671 EDD0 58                CLI
```

```
2672  EDD1  60                    RTS
2673                        ;
2674                        ;
2675                        ;
2676                        ;
2677                        ;
2678                        ;
2679                        ;
2680                        ;       'VCOUNT' INTERVAL TIMER MEASUREMENT -- TO -- POKEY FREQ REG VALU
2681                        ;               CONVERSION TABLE
2682                        ;
2683                        ;
2684                        ;               THE VALUES STORED IN THE TABLE ARE 'AUDF+7'.
2685                        ;
2686                        ; THE FOLLOWING FORMULAS WERE USED TO DETERMINE THE TABLE VALUES:
2687                        ;
2688                        ; F OUT= F IN/(2*(AUDF+M)) , WHERE F IN=1.78979 MHZ. & M=7
2689                        ;
2690                        ; FROM THIS WAS DERIVED THE FORMULA USED THE COMPUTE THE
2691                        ; TABLE VALUES BASED ON A MEASUREMENT OF THE PERIOD BY
2692                        ; AN INTERVAL OF THE 'VCOUNT' TIMER.
2693                        ;
2694                        ; AUDF+7=(11.365167)*T OUT, WHERE T OUT=# OF COUNTS (127 USEC.
2695                        ; RESOLUTION) OF 'VCOUNT' FOR 1
2696                        ; CHARACTER TIME (10 BIT TIMES).
2697                        ;
2698                        ;
2699                        ;
2700                        ;
2701                        ;               AUDF+7          ;BAUD RATE      VCOUNT INTERVAL
2702                        ;               ------          ;----------     ---------------
2703                        ;       .WORD   $27C            ;1407                   56
2704                        ;       .WORD   $2D7            ;1231                   64
2705                        ;       .WORD   $332            ;1094                   72
2706                        ;       .WORD   $38D            ;985                    80
2707  EDD2  E8 03           POKTAB: .WORD   $3E8            ;895                    88
2708  EDD4  43 04                   .WORD   $443            ;820                    96
2709  EDD6  9E 04                   .WORD   $49E            ;757                    104
2710  EDD8  F9 04                   .WORD   $4F9            ;703                    112
2711  EDDA  54 05                   .WORD   $554            ;656                    120
2712  EDDC  AF 05                   .WORD   $5AF            ;615                    128
2713  EDDE  0A 06                   .WORD   $60A            ;579                    136
2714  EDE0  65 06                   .WORD   $665            ;547                    144
2715  EDE2  C0 06                   .WORD   $6C0            ;518                    152
2716  EDE4  1A 07                   .WORD   $71A            ;492                    160
2717  EDE6  75 07                   .WORD   $775            ;469                    168
2718  EDE8  D0 07                   .WORD   $7D0            ;447                    176
2719                        ;       .WORD   $82B            ;428                    184
2720                        ;       .WORD   $886            ;410                    192
2721                        ;       .WORD   $8E1            ;394                    200
2722                        ;       .WORD   $93C            ;379                    208
2723                        ;       .WORD   $997            ;365                    216
2724                        ;       .WORD   $9F2            ;352                    224
2725                        ;       .WORD   $A4D            ;339                    232
```

```
2726                    ;       .WORD   $AA8            ;32B            240
2727                    ;       .WORD   $B03            ;31B            248
2728                    ;
2729                    ;
2730                    ;
2731                    ;
2732                    ;*****************************************************************
2733  EDEA              CRNTP3  =*
2734                            *=$14
2735  0014  00          SIOSPR: .BYTE   DSKORG-CRNTP3 ;^GSIOL IS TOO LONG
```

```
2736
2737                                    TITLE  'DISK ***** DISKP.SRC  ***** 3/9/79 ***** 4:00:00 P.M.
2738                            ;
2739                            ;
2740                            ;
2741                            ;
2742                            ;
2743                            ;
2744  0002                      STATVH  =       DVSTAT/256
2745  00EA                      STATVL  =       (-256)*STATVH+DVSTAT ;STATUS POINTER
2746                            ;
2747                            ;
2748                            ;
2749                            ;
2750                            ;       CONSTANT EQUATES
2751                            ;
2752  0031                      DISKID  =       $31             ;SERIAL BUS DISK I.D.
2753  0050                      PUTSEC  =       $50             ;DISK PUT SECTOR DCB COMMAND
2754  0053                      STATC   =       $53             ;DISK STATUS DCB COMMAND
2755  0021                      FOMAT   =       $21             ;DISK FORMAT DCB COMMAND !!!!! *****
2756  0000                      NODAT   =       0               ;SIO COMMAND FOR "NO DATA" OPERATION
2757  0040                      GETDAT  =       $40             ;SIO COMMAND FOR "DATA FROM DEVICE"
2758  0080                      PUTDAT  =       $80             ;SIO COMMAND FOR "DATA TO DEVICE"
2759                            ;
2760                            ;
2761                            ;       VECTORS
2762                            ;
2763                                    *=$E450
2764                            ;
2765  E450  4C EA ED                    JMP     DINIT           ;DISK INIT. VECTOR
2766  E453  4C F0 ED                    JMP     DSKIF           ;DISK INTERFACE ENTRY POINT
2767                            ;
2768                            ;
2769                            ;
2770                            ;
2771                            ;
2772                            ;
2773                            ;       CONSTANTS
2774                            ;
2775                                    *=DSKORG
2776                            ;
2777                            ;
2778                            ;
2779                            ;
2780                            ;
2781                            ;
2782                            ;
2783                            ;
2784                            ;
2785                            ;*****************************************************************************
2786                            ;       DISK INTERFACE ROUTINE STARTS HERE
2787                            ;*****************************************************************************
2788                            ;
2789                            ;
```

```
2790                                ;
2791                                ;
2792                                ;       DISK INTERFACE INITIALIZATION ROUTINE
2793                                ;
2794  EDEA  A9 A0           DINIT:  LDA     #160
2795  EDEC  8D 46 02                STA     DSKTIM      ;SET INITIAL DISK TIMEOUT TO 160 SEC
2796  EDEF  60                      RTS
2797                                ;
2798                                ;
2799                                ;
2800                                ;       DISK INTERFACE ENTRY POINT
2801                                ;
2802  EDF0  A9 31           DSKIF:  LDA     #DISKID
2803  EDF2  8D 00 03                STA     DDEVIC      ;SET SERIAL BUS I.D IN DCB
2804  EDF5  AD 46 02                LDA     DSKTIM
2805  EDF8  AE 02 03                LDX     DCOMND
2806  EDFB  E0 21                   CPX     #FORMAT     ;IS COMMAND A FORMAT COMMAND?
2807  EDFD  F0 02                   BEQ     PUTDTO
2808  EDFF  A9 07                   LDA     #7          ;NO, SET TIMEOUT TO 7 SECS.
2809  EE01  8D 06 03       PUTDTO: STA     DTIMLO      ;PUT DISK TIMEOUT IN DCB
2810  EE04  A2 40                   LDX     #GETDAT     ;SET "GET DATA" COMMAND FOR SIO
2811  EE06  A0 80                   LDY     #$80        ;SET BYTE COUNT TO 128
2812  EE08  AD 02 03                LDA     DCOMND      ;READ COMMAND IN DCB
2813  EE0B  C9 57                   CMP     #WRITE      ;IS COMMAND A "PUT SECTOR" COMMAND?
2814  EE0D  D0 02                   BNE     CKSTC
2815  EE0F  A2 80                   LDX     #PUTDAT     ;YES, SET "PUT DATA" COMMAND FOR SIO
2816  EE11  C9 53           CKSTC:  CMP     #STATC      ;IS COMMAND A STATUS COMMAND?
2817  EE13  D0 0C                   BNE     PUTCNT
2818  EE15  A9 EA                   LDA     #STATVL
2819  EE17  8D 04 03                STA     DBUFLO
2820  EE1A  A9 02                   LDA     #STATVH
2821  EE1C  8D 05 03                STA     DBUFHI      ;SET BUFFER ADDR TO GLOBAL STATUS BUFFER
2822  EE1F  A0 04                   LDY     #4          ;YES, SET BYTE COUNT TO 4
2823  EE21  8E 03 03       PUTCNT: STX     DSTATS      ;PUT STATUS COMMAND FOR SIO IN DCB
2824  EE24  8C 08 03                STY     DBYTLO
2825  EE27  A9 00                   LDA     #0
2826  EE29  8D 09 03                STA     DBYTHI      ;PUT BYTE COUNT IN DCB
2827  EE2C  20 59 E4                JSR     SIOV        ;CALL SERIAL I/O.
2828  EE2F  10 01                   BPL     GOODST      ;NO ERROR
2829  EE31  60                      RTS                 ;NO, GO BACK
2830  EE32  AD 02 03        GOODST: LDA     DCOMND      ;READ THE COMMAND
2831  EE35  C9 53                   CMP     #STATC      ;WAS IT A STATUS COMMAND?
2832  EE37  D0 0A                   BNE     PUTBC
2833  EE39  20 6D EE                JSR     PUTADR      ;PUT BUFFER ADDR IN TEMP REG.
2834  EE3C  A0 02                   LDY     #2
2835  EE3E  B1 15                   LDA     (BUFADR),Y  ;READ DISK TIMEOUT VALUE BYTE OF STATUS
2836  EE40  8D 46 02                STA     DSKTIM      ;PUT IT IN DISK TIMEOUT REG.
2837  EE43  AD 02 03        PUTBC:  LDA     DCOMND
2838  EE46  C9 21                   CMP     #FORMAT     ;WAS COMMAND A FORMAT COMMAND?
2839  EE48  D0 1F                   BNE     ENDDIF
2840  EE4A  20 6D EE        FMTD:   JSR     PUTADR      ;YES, PUT BUFFER ADDR INTO TEMP REG
2841  EE4D  A0 FE                   LDY     #$FE        ;SET BUFFER POINTER
2842  EE4F  C8              TWICE:  INY
2843  EE50  C8                      INY                 ;INCR BUFFER POINTER BY 2
```

```
2844  EE51  B1 15        RDBAD:  LDA     (BUFADR),Y  ;READ LO BYTE BAD SECTOR DATA
2845  EE53  C9 FF                CMP     #$FF
2846  EE55  D0 F8                BNE     TWICE       ;IS IT "FF" ?
2847  EE57  C8                   INY                 ;YES,
2848  EE58  B1 15                LDA     (BUFADR),Y  ;READ HI BYTE BAD SECTOR DATA
2849  EE5A  C8                   INY
2850  EE5B  C9 FF                CMP     #$FF
2851  EE5D  D0 F2                BNE     RDBAD       ;IS IT "FF" ?
2852  EE5F  88                   DEY
2853  EE60  88                   DEY                 ;YES,
2854  EE61  8C 08 03             STY     DBYTLO      ;PUT BAD SECTOR BYTE COUNT INTO DCB
2855  EE64  A9 00                LDA     #0
2856  EE66  8D 09 03             STA     DBYTHI
2857  EE69  AC 03 03     ENDDIF: LDY     DSTATS
2858  EE6C  60                   RTS
2859                      ;
2860                      ;
2861                      ;
2862                      ;
2863                      ;       S U B R O U T I N E S
2864                      ;
2865                      ;
2866                      ;       PUT BUFFER ADDR FROM DCB INTO TEMP REG
2867                      ;
2868  EE6D  AD 04 03      PUTADR: LDA     DBUFLO
2869  EE70  85 15                STA     BUFADR
2870  EE72  AD 05 03             LDA     DBUFHI
2871  EE75  85 16                STA     BUFADR+1    ;PUT BUFFER ADDR IN TEMP REG
2872  EE77  60                   RTS
2873                      ;***************************************************************
2874                      ;
2875                      ;
2876                      ;       SPARE BYTE OR MODULE TOO LONG FLAG
2877                      ;
2878  EE78                CRNTP4  =       *
2879                      ;
2880                              *=$14
2881  0014  00            DSKSPR: .BYTE   PRNORG-CRNTP4 ;^GDISKP TOO LONG
2882                      ;
```

```
2883                                    .PAGE
2884                                    .TITLE  'PRINTER  *****  PRINTP.SRC  ***** 3/9/79 ***** 4:00:00
2885                            ;
2886                            ;
2887                            ;
2888                            ;
2889                            ;
2890                            ;
2891                            ;
2892                            ;
2893                            ;
2894                            ;
2895                            ;
2896                            ;       DEVICE NUMBER OR CODE EQUATES
2897                            ;
2898  0002                      OPNOUT  =       $2              ;IOCB OPEN FOR OUTPUT COMMAND
2899  0028                      NBUFSZ  =       40              ;PRINT NORMAL BUFFER SIZE
2900  0014                      DBUFSZ  =       20              ;PRINT DOUBLE BUFFFER SIZE
2901  001D                      SBUFSZ  =       29              ;PRINT SIDEWAYS BUFFER SIZE
2902  0040                      PDEVN   =       $40             ;PRINTER DEVICE NUMBER
2903  0057                      WRITEC  =       $57             ;DCB WRITE COMMAND
2904  0020                      SPACE   =       $20             ;ASCII SPACE CHAR.
2905  004E                      N       =       $4E             ;ASCII "N" CHAR.
2906  0044                      D       =       $44             ;ASCII "D" CHAR.
2907  0053                      S       =       $53             ;ASCII "S" CHAR.
2908                            ;
2909                            ;
2910                            ;       PRINTER HANDLER ENTRY POINTS
2911                            ;
2912                            ;
2913                            ;
2914                            ;
2915                                    *=$E430
2916                            ;
2917  E430  9E EE                       .WORD   PHOPEN-1        ;PRINTER HANDLER OPEN
2918  E432  DB EE                       .WORD   PHCLOS-1        ;PH CLOSE
2919  E434  9D EE                       .WORD   BADST-1         ;PH READ
2920  E436  A6 EE                       .WORD   PHWRIT-1        ;PH WRITE
2921  E438  80 EE                       .WORD   PHSTAT-1        ;PH STATUS
2922  E43A  9D EE                       .WORD   BADST-1         ;PH SPECIAL
2923  E43C  4C 7B EE                    JMP     PHINIT          ;PH INIT.
2924  E43F  00                          .BYTE   0               ;ROM FILLER
2925                            ;
2926                            ;
2927                            ;
2928                            ;
2929                            ;
2930                                    *=PRNORG
2931                            ;
2932                            ;
2933                            ;
2934                            ;
2935                            ;       PRINTER HANDLER INITIALIZATION ROUTINE
2936                            ;
```

```
2937 EE78 A9 1E        PHINIT: LDA     #30
2938 EE7A 85 1C                STA     PTIMOT      ;SET UP INITIAL PRINTER TIMEOUT OF 30 SEC.
2939 EE7C 60                   RTS
2940                   ;
2941                   ;
2942                   ;       PRINTER HANDLER CONSTANTS
2943                   ;
2944 EE7D EA 02        PHSTLO: .WORD   DVSTAT      ;STATUS BUFFER POINTER
2945 EE7F C0 03        PHCHLO: .WORD   PRNBUF      ;CHAR. BUFFER POINTER
2946                   ;
2947                   ;
2948                   ;
2949                   ;       ************************
2950                   ;       PRINTER HANDLER ROUTINES
2951                   ;       ************************
2952                   ;
2953                   ;
2954                   ;
2955                   ;
2956                   ;
2957                   ;       PRINTER HANDLER STATUS ROUTINE
2958                   ;
2959 EE81 A9 04        PHSTAT: LDA     #4
2960 EE83 85 1E                STA     PBUFSZ      ;SET BUFFER SIZE TO 4 BYTES
2961 EE85 AE 7D EE             LDX     PHSTLO
2962 EE88 AC 7E EE             LDY     PHSTLO+1    ;SET POINTER TO STATUS BUFFER
2963 EE8B A9 53                LDA     #STATC      ;SET COMMAND TO "STATUS"
2964 EE8D 8D 02 03             STA     DCOMND      ;SET STATUS COMMAND
2965 EE90 8D 0A 03             STA     DAUX1
2966 EE93 20 E6 EE             JSR     SETDCB      ;GO SETUP DCB
2967 EE96 20 59 E4             JSR     SIOV        ;SEND STATUS COMMAND
2968 EE99 30 03                BMI     BADST       ;GO IF ERROR
2969 EE9B 20 14 EF             JSR     PHPUT       ;YES, PUT STATUS INTO GLOBAL BUFFER.
2970 EE9E 60           BADST:  RTS
2971                   ;
2972                   ;
2973                   ;
2974                   ;
2975                   ;       PRINTER HANDLER OPEN ROUTINE
2976                   ;
2977 EE9F 20 81 EE     PHOPEN: JSR     PHSTAT      ;DO STATUS COMMAND TO SIO
2978 EEA2 A9 00                LDA     #0
2979 EEA4 85 1D                STA     PBPNT       ;CLEAR PRINT BUFFER POINTER
2980 EEA6 60                   RTS
2981                   ;
2982                   ;
2983                   ;
2984                   ;
2985                   ;       PRINTER HANDLER WRITE ROUTINE
2986                   ;
2987 EEA7 85 1F        PHWRIT: STA     PTEMP       ;SAVE ACCUM
2988 EEA9 20 1A EF             JSR     PRMODE      ;GO DETERMINE PRINT MODE
2989 EEAC A6 1D                LDX     PBPNT
2990 EEAE A5 1F                LDA     PTEMP       ;GET CHAR. SENT BY CIO
```

```
2991  EEB0  9D C0 03          STA     PRNBUF,X        ;PUT CHAR. IN PRINT BUFFER
2992  EEB3  E8                INX                     ;INCR. BUFFER POINTER
2993  EEB4  E4 1E             CPX     PBUFSZ          ;BUFFER POINTER=BUFFER SIZE?
2994  EEB6  F0 13             BEQ     BUFFUL
2995  EEB8  86 1D             STX     PBPNT           ;SAVE BUFFER POINTER
2996  EEBA  C9 9B             CMP     #CR             ;IS CHAR. = EOL ?
2997  EEBC  F0 03             BEQ     BLFILL          ;IF YES, GO DO BLANK FILL.
2998  EEBE  A0 01             LDY     #SUCCES         ;PUT GOOD STATUS IN Y REG FOR CIO.
2999  EEC0  60                RTS
3000  EEC1  A9 20       BLFILL: LDA   #SPACE          ;PUT BLANK IN ACCUM.
3001  EEC3  9D C0 03    FILLBF: STA   PRNBUF,X        ;STORE IT IN PRINT BUFFER.
3002  EEC6  E8                INX
3003  EEC7  E4 1E             CPX     PBUFSZ
3004  EEC9  D0 F8             BNE     FILLBF          ;BUFFER BLANK FILLED?
3005  EECB  A9 00       BUFFUL: LDA   #0
3006  EECD  85 1D             STA     PBPNT           ;CLEAR PRINT BUFFER POINTER
3007  EECF  AE 7F EE          LDX     PHCHLO
3008  EED2  AC 80 EE          LDY     PHCHLO+1        ;SET POINTER TO PRINT BUFFER
3009  EED5  20 E6 EE          JSR     SETDCB          ;GO SETUP DCB
3010  EED8  20 59 E4          JSR     SIOV            ;SEND PRINT COMMAND
3011  EEDB  60                RTS                     ;YES.
3012                    ;
3013                    ;
3014                    ;
3015                    ;
3016                    ;       PRINTER HANDLER CLOSE ROUTINE
3017                    ;
3018  EEDC  20 1A EF    PHCLOS: JSR   PRMODE          ;GO DETERMINE PRINT MODE
3019  EEDF  A6 1D             LDX     PBPNT
3020  EEE1  D0 DE             BNE     BLFILL
3021  EEE3  A0 01             LDY     #SUCCES
3022  EEE5  60                RTS
3023                    ;
3024                    ;
3025                    ;
3026                    ;
3027                    ;
3028
3029
3030                    ,
3031                    ;       S U B R O U T I N E S
3032                    ;
3033                    ;
3034                    ;
3035                    ;
3036                    ;
3037                    ;       SET UP DCB TO CALL SIO
3038                    ;
3039  EEE6  8E 04 03    SETDCB: STX   DBUFLO
3040  EEE9  8C 05 03          STY     DBUFHI          ;SET BUFFER POINTER
3041  EEEC  A9 40             LDA     #PDEVN
3042  EEEE  8D 00 03          STA     DDEVIC          ;SET PRINTER BUS I.D. FOR DCB
3043  EEF1  A9 01             LDA     #1
3044  EEF3  8D 01 03          STA     DUNIT           ;SET UNIT NUMBER TO 1
```

```
3045 EEF6 A9 80          LDA   #$80        ;DEVICE WILL EXPECT DATA
3046 EEF8 AE 02 03       LDX   DCOMND
3047 EEFB E0 53          CPX   #STATC      ;STATUS COMMAND?
3048 EEFD D0 02          BNE   PSIOC
3049 EEFF A9 40          LDA   #$40        ;EXPECT DATA FROM DEVICE
3050 EF01 8D 03 03 PSIOC: STA   DSTATS     ;SET SIO MODE COMMAND.
3051 EF04 A5 1E          LDA   PBUFSZ
3052 EF06 8D 08 03       STA   DBYTLO      ;SET LO BYTE COUNT
3053 EF09 A9 00          LDA   #0
3054 EF0B 8D 09 03       STA   DBYTHI      ;SET HI BYTE COUNT
3055 EF0E A5 1C          LDA   PTIMDT
3056 EF10 8D 06 03       STA   DTIMLO      ;SET DEVICE TIMEOUT COUNT
3057 EF13 60             RTS
3058                     ;
3059                     ;
3060                     ;
3061                     ;
3062                     ;       GET DEVICE TIMEOUT FROM STATUS & SAVE IT
3063                     ;
3064 EF14 AD EC 02 PHPUT: LDA   DVSTAT+2
3065 EF17 85 1C          STA   PTIMOT      ;SAVE DEVICE TIMEOUT
3066 EF19 60             RTS
3067                     ;
3068                     ;
3069                     ;
3070                     ;
3071                     ;DETERMINE PRINT MODE & SETUP PRINT BUFFER SIZE, DCB PRINT
3072                     ;COMMAND, & DCB AUX1 FOR PRINT MODE
3073                     ;
3074 EF1A A0 57 PRMODE: LDY   #WRITEC     ;PUT WRITE COMMAND IN Y REG
3075 EF1C A5 2B          LDA   ICAX2Z      ;READ PRINT MODE
3076 EF1E C9 4E CMODE:  CMP   #N
3077 EF20 D0 04          BNE   CDUBL       ;PRINT NORMAL ?
3078 EF22 A2 28          LDX   #NBUFSZ     ;YES, SET NORMAL CHAR. BUFFER SIZE
3079 EF24 D0 0E          BNE   SETBSZ
3080 EF26 C9 44 CDUBL:   CMP   #D
3081 EF28 D0 04          BNE   CSIDE       ;PRINT DOUBLE?
3082 EF2A A2 14          LDX   #DBUFSZ     ;YES, SET DOUBLE CHAR. BUFFER SIZE
3083 EF2C D0 06          BNE   SETBSZ
3084 EF2E C9 53 CSIDE:   CMP   #S          ;PRINT SIDEWAYS ?
3085 EF30 D0 0B          BNE   GOERR       ;IF NOT, GO TO ERROR ROUTINE
3086 EF32 A2 1D          LDX   #SBUFSZ     ;YES, SET SIDEWAYS BUFFER SIZE
3087 EF34 86 1E SETBSZ:  STX   PBUFSZ      ;STORE PRINT BUFFER SIZE
3088 EF36 8C 02 03       STY   DCOMND      ;STORE DCB COMMAND
3089 EF39 8D 0A 03       STA   DAUX1       ;STORE DCB AUX1 PRINT MODE
3090 EF3C 60             RTS
3091 EF3D A9 4E GOERR:   LDA   #N          ;SET DEFAULT PRINT MODE TO NORMAL
3092 EF3F D0 DD          BNE   CMODE
3093                     ;*************************************************************
3094                     ;
3095                     ;
3096                     ;       SPARE BYTE OR MODULE TOO LONG FLAG
3097                     ;
3098 EF41 CRNTP5 =       *
```

```
3099                    ;
3100                            *=$14
3101                    ;
3102  0014  00         PRNSPR: .BYTE   CASORG-CRNTP5 ;^GPRINTP TOO LONG
3103                    ;
```

```
3104                                    .PAGE
3105                                    LIST    S
3106                                    .TITLE  'CASSET HANDLER 3/12 (DK1:CASCV)'
3107  0003                      CBUFH   =       CASBUF/256
3108  00FD                      CBUFL   =       (-256)*CBUFH+CASBUF
3109  0040                      SRSTA   =       $40             ;SIO READ STATUS
3110  0080                      SWSTA   =       $80             ;SIO WRITE STATUS
3111                            ;
3112                            ;
3113  00FC                      DTA     =       $FC             ;DATA RECORD TYPE BYTE
3114  00FA                      DT1     =       $FA             ;LAST DATA RECORD
3115  00FE                      EOT     =       $FE             ;END OF TAPE
3116  00FB                      HDR     =       $FB             ;HEADER
3117  0002                      TONE1   =       2               ;CHANGE TO RECORD MODE TONE
3118  0001                      TONE2   =       1               ;PRESS PLAY TONE
3119                            ;
3120                            ;
3121                            ;
3122                                    *=CASETV
3123  E440  4B EF 2A F0                 .WORD   OPENC-1,CLOSEC-1,GBYTE-1,PBYTE-1,STATU-1,SPECIAL-1
3124  E444  D5 EF 0F F0
3125  E448  27 F0 4A EF
3126  E44C  4C 41 EF                    JMP     INIT
3127  E44F  00                          .BYTE   0               ;ROM FILLER BYTE
3128                            ;
3129                            ;
3130                            ;
3131                            ; USED IN MONITP FOR CASSETTE BOOT
3132                            ;
3133                                    *=RBLOKV
3134  E47A  4C E9 EF                    JMP     RBLOK
3135                            ;
3136                                    *=CSOPIV
3137  E47D  4C 5D EF                    JMP     OPINP
3138                            ;
3139                            ;
3140                                    *=CASORG
3141                            ;
3142                            ;
3143                            ; INIT ROUTINE
3144                            ;
3145  EF41  A9 CC               INIT:   LDA     #$CC
3146  EF43  8D EE 02                    STA     CBAUDL
3147  EF46  A9 05                       LDA     #$05
3148  EF48  8D EF 02                    STA     CBAUDH          ;SET CASSET BAUD RATE TO 600
3149  EF4B  60                  SPECIAL:RTS                     ;THATS ALL FOLKS
```

```
3150                                    .PAGE
3151                                    ;
3152                                    ; OPEN FUNCTION - WITH NO TIMING ADJUST
3153                                    ;
3154  EF4C A5 2B            OPENC:  LDA     ICAX2Z          ;GET AX2
3155  EF4E 85 3E                    STA     FTYPE           ;SAVE IT FOR FUTURE REFERENCE
3156  EF50 A5 2A                    LDA     ICAX1Z
3157  EF52 29 0C                    AND     #$0C            ;IN AND OUT BITS
3158  EF54 C9 04                    CMP     #$04
3159  EF56 F0 05                    BEQ     OPINP
3160  EF58 C9 08                    CMP     #$08            ;SEE IF OPEN FOR OUTPUT
3161  EF5A F0 39                    BEQ     OPOUT
3162  EF5C 60                       RTS                     ;IF ALREADY OPEN, RETURN LEAVING STATUS=$84
3163  EF5D A9 00            OPINP:  LDA     #0
3164  EF5F 8D 89 02                 STA     WMODE           ;SET READ MODE
3165  EF62 85 3F                    STA     FEOF            ;NO EOF YET
3166  EF64 A9 01            SFH:    LDA     #TONE2          ;TONE FOR PRESS PLAY
3167  EF66 20 58 F0                 JSR     BEEP            ;GO BEEP
3168  EF69 30 24                    BMI     OPNRTN          ;IF ERROR DURING BEEP
3169  EF6B A9 34                    LDA     #MOTRGO
3170  EF6D 8D 02 D3                 STA     PACTL           ;TURN MOTOR ON
3171  EF70 A0 40                    LDY     #$40            ;5-31-79 - 9 SEC READ LEADER
3172  EF72 A2 02                    LDX     #2
3173  EF74 A9 03                    LDA     #3
3174  EF76 8D 2A 02                 STA     CDTMF3
3175  EF79 20 5C E4                 JSR     SETVBV          ;SET UP VBLANK TIMER
3176  EF7C AD 2A 02         WAITTM: LDA     CDTMF3
3177  EF7F D0 FB                    BNE     WAITTM          ;WAIT FOR MOTOR TO COME UP TO SPEED
3178  EF81 A9 80                    LDA     #$80            ;NEXT BYTE=NO BYTES IN BUFFER
3179  EF83 85 3D                    STA     BPTR
3180  EF85 8D 8A 02                 STA     BLIM
3181  EF88 4C D3 EF                 JMP     OPOK            ;OPEN OK
3182                                    ;
3183                                    ; OPEN FOR OUTPUT
3184                                    ;
3185  EF8B A0 80            PBRK:   LDY     #BRKABT         ;BREAK KEY ABORT STATUS
3186  EF8D C6 11                    DEC     BRKKEY          ;RESET BREAK KEY
3187  EF8F A9 00            OPNRTN: LDA     #0              ;CLEAR WRITE MODE FLAG
3188  EF91 8D 89 02                 STA     WMODE
3189  EF94 60                       RTS                     ;AND EXIT.
3190                                    ;
3191  EF95 A9 80            OPOUT:  LDA     #$80
3192  EF97 8D 89 02                 STA     WMODE           ;SET WRITE MODE
3193  EF9A A9 02                    LDA     #TONE1          ;TELL USER TO TURN ON RECORD MODE
3194  EF9C 20 58 F0                 JSR     BEEP
3195  EF9F 30 EE                    BMI     OPNRTN          ;IF ERROR DURING BEEP
3196  EFA1 A9 CC                    LDA     #$CC            ;SET BAUD RATE
3197  EFA3 8D 04 D2                 STA     AUDF3           ;WHICH SEEMS TO BE NESSECARY
3198  EFA6 A9 05                    LDA     #$05            ;FOR SOME OBSCURE REASON
3199  EFA8 8D 06 D2                 STA     AUDF4
3200  EFAB A9 60                    LDA     #$60
3201  EFAD 8D 00 03                 STA     DDEVIC
3202  EFB0 20 68 E4                 JSR     SENDEV          ;TELL POKEY TO WRITE MARKS
3203  EFB3 A9 34                    LDA     #MOTRGO         ;WRITE 5 SEC BLANK TAPE
```

```
3204  EFB5  8D 02 D3          STA     PACTL
3205  EFB8  A9 03             LDA     #3
3206  EFBA  A2 04             LDX     #4          ;5/30/79 20 SEC LEADER
3207  EFBC  A0 80             LDY     #$80
3208  EFBE  20 5C E4          JSR     SETVBV
3209  EFC1  A9 FF             LDA     #$FF
3210  EFC3  8D 2A 02          STA     CDTMF3
3211  EFC6  A5 11       WDLR: LDA     BRKKEY
3212  EFC8  F0 C1             BEQ     PBRK        ;IF BREAK DURING WRITE LEADER
3213  EFCA  AD 2A 02          LDA     CDTMF3
3214  EFCD  D0 F7             BNE     WDLR
3215  EFCF  A9 00             LDA     #0          ;INIT BUFFER POINTER
3216  EFD1  85 3D             STA     BPTR
3217  EFD3  A0 01       OPOK: LDY     #SUCCES
3218  EFD5  60                RTS
```

```
3219                            .PAGE
3220                    ;
3221                    ; GET BYTE
3222                    ;
3223  EFD6  A5 3F       GBYTE:  LDA     FEOF            ;IF AT EOF ALREADY
3224  EFD8  30 33               BMI     ISEOF           ;RETURN EOF STATUS
3225  EFDA  A6 3D               LDX     BPTR            ;BUFFER POINTER
3226  EFDC  EC 8A 02            CPX     BLIM            ;IF END OF BUFFER
3227  EFDF  F0 08               BEQ     RBLOK           ;READ ANOTHER BLOCK
3228  EFE1  BD 00 04            LDA     CASBUF+3,X      ;GET NEXT BYTE
3229  EFE4  E6 3D               INC     BPTR            ;BUMP POINTER
3230  EFE6  A0 01               LDY     #SUCCES         ;OK STATUS
3231  EFE8  60          GBX:    RTS
3232  EFE9  A9 52       RBLOK:  LDA     #'R             ;READ OPCODE
3233  EFEB  20 95 F0            JSR     SIOSB           ;SIO ON SYS BUF
3234  EFEE  98                  TYA
3235  EFEF  30 F7               BMI     GBX             ;IF SIO ERRORS, RETURN
3236  EFF1  A9 00               LDA     #0
3237  EFF3  85 3D               STA     BPTR            ;RESET POINTER
3238  EFF5  A2 80               LDX     #$80            ;DEFAULT # BYTES
3239  EFF7  AD FF 03            LDA     CASBUF+2
3240  EFFA  C9 FE               CMP     #EOT
3241  EFFC  F0 0D               BEQ     ATEOF           ;IF HEADER, GO READ AGAIN
3242  EFFE  C9 FA               CMP     #DT1            ;IF LAST DATA REC
3243  F000  D0 03               BNE     NLR
3244  F002  AE 7F 04            LDX     CASBUF+130      ;LAST DATA RECORD, GET # BYTES
3245  F005  8E 8A 02    NLR:    STX     BLIM
3246  F008  4C D6 EF            JMP     GBYTE           ;GET NEXT BYTE
3247  F00B  C6 3F       ATEOF:  DEC     FEOF            ;SET FEOF
3248  F00D  A0 88       ISEOF:  LDY     #EOFERR         ;ENDFILE STATUS
3249  F00F  60                  RTS
```

```
3250                            .PAGE
3251                    ;
3252                    ; PUT BYTE TO BUFFER
3253                    ;
3254  F010  A6 3D      PBYTE:  LDX     BPTR            ;BUFFER POINTER
3255  F012  9D 00 04          STA     CASBUF+3,X      ;STORE CHAR AWAY
3256  F015  E6 3D             INC     BPTR            ;BUMP POINTER
3257  F017  A0 01             LDY     #SUCCES         ;OK STATUS
3258  F019  E0 7F             CPX     #127            ;IF BUFFER FULL
3259  F01B  F0 01             BEQ     *+3
3260  F01D  60                RTS
3261                    ; WRITE OUT THE BUFFER
3262  F01E  A9 FC             LDA     #DTA            ;RECORD TYPE = DATA
3263  F020  20 D2 F0          JSR     WSIOSB          ;DO WRITE ON SYSTEM BUFFER
3264  F023  A9 00             LDA     #0
3265  F025  85 3D             STA     BPTR            ;RESET BUFFER POINTER
3266  F027  60                RTS                     ;EXIT.
```

```
3267                            .PAGE
3268                    ;
3269                    ; STATUS - RETURN STATUS INFO THRU DVSTAT
3270                    ;
3271  F028  A0 01      STATU:  LDY     #SUCCES
3272  F02A  60              RTS
```

```
3273                                    .PAGE
3274                            ;
3275                            ; CLOSE
3276                            ;
3277  F02B  AD 89 02    CLOSEC: LDA     WMODE           ;SEE IF WRITING
3278  F02E  30 08               BMI     CLWRT           ;GO CLOSE FOR WRITE
3279                            ;CLOSE FOR READ - FLAG CLOSED
3280  F030  A0 01               LDY     #SUCCES         ;SUCCESSFULL
3281  F032  A9 3C       FCAX:   LDA     #MOTRST         ;STOP THE MOTOR IN CASE WAS SHORT IRG MODE
3282  F034  8D 02 D3            STA     PACTL
3283  F037  60                  RTS
3284  F038  A6 3D       CLWRT:  LDX     BPTR            ;BUFFER POINTER
3285  F03A  F0 0A               BEQ     WTLR            ;IF NO DATA BYTES IN BUFFER, NO DT1 REC
3286  F03C  8E 7F 04            STX     CASBUF+130      ;WRITE TO LAST RECORD
3287  F03F  A9 FA               LDA     #DT1            ;REC TYPE
3288  F041  20 D2 F0            JSR     WSIOSB          ;WRITE OUT USER BUFFER
3289  F044  30 EC               BMI     FCAX            ;GO IF ERROR
3290  F046  A2 7F       WTLR:   LDX     #127            ;ZERO BUFFER
3291  F048  A9 00               LDA     #0
3292  F04A  9D 00 04     ZTBUF: STA     CASBUF+3,X
3293  F04D  CA                  DEX
3294  F04E  10 FA               BPL     ZTBUF
3295  F050  A9 FE               LDA     #EOT            ;WRITE EOT RECORD
3296  F052  20 D2 F0            JSR     WSIOSB
3297  F055  4C 32 F0            JMP     FCAX            ;FLAG CLOSED AND EXIT
```

```
3298                                    PAGE
3299                            ;
3300                            ; SUBROUTINES
3301                            ;
3302                            ; BEEP - GENERATE TONE ON KEYBOARD SPEAKER
3303                            ; ON ENTRY A= FREQ
3304                            ;
3305  F058  85 40               BEEP:   STA     FREQ
3306  F05A  A5 14               BEEP1:  LDA     RTCLOK+2        ;CURRENT CLOCK
3307  F05C  18                          CLC
3308  F05D  69 1E                       ADC     #30             ;1 SEC TONE
3309  F05F  AA                          TAX
3310  F060  A9 FF               WFL:    LDA     #$FF
3311  F062  8D 1F D0                    STA     CONSOL          ;TURN ON SPEAKER
3312  F065  A9 00                       LDA     #0
3313  F067  A0 F0                       LDY     #$F0
3314  F069  88                          DEY
3315  F06A  D0 FD                       BNE     *-1
3316  F06C  8D 1F D0                    STA     CONSOL          ;TURN OFF SPEAKER
3317  F06F  A0 F0                       LDY     #$F0
3318  F071  88                          DEY
3319  F072  D0 FD                       BNE     *-1
3320  F074  E4 14                       CPX     RTCLOK+2        ;SEE IF 1 SEC IS UP YET
3321  F076  D0 E8                       BNE     WFL
3322  F078  C6 40                       DEC     FREQ            ;COUNT BEEPS
3323  F07A  F0 0B                       BEQ     WFAK            ;IF ALL DONE GO WAIT FOR KEY
3324  F07C  8A                          TXA
3325  F07D  18                          CLC
3326  F07E  69 0A                       ADC     #10
3327  F080  AA                          TAX
3328  F081  E4 14                       CPX     RTCLOK+2
3329  F083  D0 FC                       BNE     *-2
3330  F085  F0 D3                       BEQ     BEEP1           ;UNCOND GO BEEP AGIN
3331  F087  20 8C F0            WFAK:   JSR     WFAK1           ;USE SIMULATED "JMP (KGETCH)"
3332  F08A  98                          TYA
3333  F08B  60                          RTS
3334  F08C  AD 25 E4            WFAK1:  LDA     KEYBDV+5
3335  F08F  48                          PHA
3336  F090  AD 24 E4                    LDA     KEYBDV+4        ;SIMULATE "JMP (KGETCH)"
3337  F093  48                          PHA
3338  F094  60                          RTS
3339                            ;
3340                            ; SIOSB - CALL SIO ON SYSTEM BUFFER
3341                            ;
3342  F095  8D 02 03            SIOSB:  STA     DCOMND          ;SAVE COMMAND
3343  F098  A9 00                       LDA     #0
3344  F09A  8D 09 03                    STA     DBYTHI          ;SET BUFFER LENGTH
3345  F09D  A9 83                       LDA     #131
3346  F09F  8D 08 03                    STA     DBYTLO
3347  F0A2  A9 03                       LDA     #CBUFH
3348  F0A4  8D 05 03                    STA     DBUFHI          ;SET BUFFER ADDRESS
3349  F0A7  A9 FD                       LDA     #CBUFL
3350  F0A9  8D 04 03                    STA     DBUFLO
3351  F0AC  A9 60               CSIO:   LDA     #$60            ;CASSET PSEUDO DEVICE
```

```
3352  F0AE  8D 00 03          STA     DDEVIC
3353  F0B1  A9 00             LDA     #0
3354  F0B3  8D 01 03          STA     DUNIT
3355  F0B6  A9 23             LDA     #35             ;DEVICE TIMEOUT (5/30/79)
3356  F0B8  8D 06 03          STA     DTIMLO
3357  F0BB  AD 02 03          LDA     DCOMND          ;GET COMMAND BACK
3358  F0BE  A0 40             LDY     #SRSTA          ;SIO READ STATUS COMMAND
3359  F0C0  C9 52             CMP     #'R
3360  F0C2  F0 02             BEQ     *+4
3361  F0C4  A0 80             LDY     #SWSTA          ;SIO WRITE STATUS COMMAND
3362  F0C6  8C 03 03          STY     DSTATS          ;SET STATUS FOR SIO
3363  F0C9  A5 3E             LDA     FTYPE
3364  F0CB  8D 0B 03          STA     DAUX2           ;INDICATE IF SHORT IRG MODE
3365  F0CE  20 59 E4          JSR     SIOV            ;GO CALL SIO
3366  F0D1  60                RTS
3367                        ;
3368                        ; WSIOSB - WRITE SIO SYSTEM BUFFER
3369                        ;
3370  F0D2  8D FF 03        WSIOSB: STA     CASBUF+2        ;STORE TYPE BYTE
3371  F0D5  A9 55             LDA     #$55
3372  F0D7  8D FD 03          STA     CASBUF+0
3373  F0DA  8D FE 03          STA     CASBUF+1
3374  F0DD  A9 57             LDA     #'W             ;WRITE
3375  F0DF  20 95 F0          JSR     SIOSB           ;CALL SIO ON SYSTEM BUFFER
3376  F0E2  60                RTS     AND             ;RETURN
3377  F0E3                  CRNTP6  =*
3378                          *=$14
3379  0014  00              CASSPR: .BYTE   MONORG-CRNTP6 ;^GCASCV IS TOO LONG
```

```
3380
3381                            .TITLE  'MONITOR  *****  MONITP.SRC  ***** 3/9/79  *****  4:00:0
3382                    ;
3383                    ;
3384                    ;
3385                    ;       CONSTANT EQUATES
3386                    ;
3387  0009              PUTTXT  =       $9              ;"PUT TEXT RECORD" CIO COMMAND CODE
3388  0007              GETCAR  =       $7              ;"GET CHARACTER" CIO COMMAND CODE
3389  000B              PUTCAR  =       $B              ;"PUT CHARACTER" CIO COMMAND CODE
3390  0000              INIMLL  =       $00             ;INITIAL MEM LO LOW BYTE
3391  0007              INIMLH  =       $07             ;INITIAL MEM LO HIGH BYTE
3392  0000              SEX     =       $0              ;SCREEN EDITOR IOCB INDEX
3393  007D              CLS     =       $7D             ;CLEAR SCREEN CODE
3394  0092              CTRLC   =       $92             ;KEYBOARD CODE FOR 'CONTROL C'
3395  0088              EOF     =       $88             ;CASSETTE END OF FILE CODE
3396  0000              LIRG    =       $0              ;LONG IRG TYPE CODE
3397                    ;
3398  0004              BUFFH   =       (CASBUF+3)/256
3399  0000              BUFFL   =       (-256)*BUFFH+CASBUF+3 ;BUFFER POINTER
3400                    ;
3401                    ;
3402                    ;
3403                    ;THE FOLLOWING EQUATES ARE IN THE CARTRIDGE ADDRESS SPACE.
3404                    ;
3405                    ;
3406                    ;       "B" CARTRIDGE ADDR'S ARE 8000-9FFF (36K CONFIG. ONLY)
3407                    ;       "A" CART. ADDR'S ARE A000-BFFF (36K CONFIG. ONLY)
3408                    ;
3409                    ;       "A" CART. ADDR'S ARE B000-BFFF (48K CONFIG. ONLY)
3410                    ;
3411                            *=$BFFA
3412  BFFA              CARTCS: .RES    2               ;CARTRIDGE COLD START ADDRESS.
3413  BFFC              CART:   .RES    1               ;CARTRIDGE AVAILABLE FLAG BYTE.
3414  BFFD              CARTFG: .RES    1               ;CARTRIDGE FLAG BYTE.  BIT 0=FLAG1,
3415  BFFE              CARTAD: .RES    2               ;2-BYTE CARTRIDGE START VECTOR
3416                    ;
3417                    ;
3418                    ;       CARTRIDGE FLAG ACTION DEFINITIONS
3419                    ;
3420                    ;
3421                    ;       BIT             ACTION IF SET
3422                    ;
3423                    ;       7               SPECIAL -- DON'T POWER-UP, JUST RUN CARTRIDGE
3424                    ;       6-3             NONE
3425                    ;       2               RUN CARTRIDGE
3426                    ;       1               NONE
3427                    ;       0               BOOT DOS
3428                    ;
3429                    ;
3430                    ;       *****
3431                    ;       NOTE
3432                    ;       *****
3433                    ;
```

```
3434                            ;       1. IF BIT2 IS 0, GOTO BLACKBOARD MODE.
3435                            ;       2. IF BIT0 SET, THE DISK WILL BE BOOTED BEFORE ANY
3436                            ;          OTHER ACTION.
3437                            ;
3438                            ;
3439                            ;
3440                            ;
3441                            ;
3442                            ;
3443                            ;
3444                            ;
3445                            ;
3446                            ;       POWER-UP VECTOR
3447                            ;
3448                            ;*****************************
3449                            ;       * = $FFFC
3450                            ;
3451                            ;PVECT  .WORD   PWRUP           ;POWER-UP VECTOR
3452                            ;*************************
3453                            ;
3454                            ;
3455                            ;
3456                            ;
3457                            ;
3458                            ;       ENTRY POINT VECTOR
3459                            ;
3460                                    *=BLKBDV
3461                            ;
3462  E471  4C 23 F2                    JMP     SIGNON      ;BLACK BOARD VECTOR
3463                            ;
3464                                    *=WARMSV
3465                            ;
3466  E474  4C 1B F1                    JMP     RESET       ;WARM START VECTOR
3467                            ;
3468                                    *=COLDSV
3469                            ;
3470  E477  4C 25 F1                    JMP     PWRUP       ;COLD START VECTOR (9000 FOR RAM VECTOR WRIT
3471                            ;
3472                                    *=$9000
3473  9000  20 0C 90                    JSR     $900C
3474  9003  4C 25 F1                    JMP     PWRUP       ;(TO HANDLE RAM VECTOR WRITING)
3475  9006  20 0C 90                    JSR     $900C
3476  9009  4C 1B F1                    JMP     RESET
3477                            ;
3478                            ;
3479                            ;
3480                                    *=MONORG
3481                            ;
3482                            ;
3483                            ;
3484                            ;
3485                            ;       HANDLER TABLE ENTRIES
3486                            ;
3487  F0E3  50                  TBLENT: .BYTE   'P'
```

```
3488  F0E4  30 E4              .WORD   PRINTV
3489  F0E6  43                 .BYTE   'C'
3490  F0E7  40 E4              .WORD   CASETV
3491  F0E9  45                 .BYTE   'E'
3492  F0EA  00 E4              .WORD   EDITRV
3493  F0EC  53                 .BYTE   'S'
3494  F0ED  10 E4              .WORD   SCRENV
3495  F0EF  4B                 .BYTE   'K'
3496  F0F0  20 E4              .WORD   KEYBDV
3497                        ;
3498                        ;
3499                        ;TBLLEN  =       IDENT-TBLENT-1  HANDLER TABLE LENGTH   "MOVED TO LINE 8
3500                        ;
3501                        ;       ***** PRINT MESSAGES *****
3502                        ;
3503                        ;
3504  F0F2  7D 41 54 41     IDENT:  .BYTE   CLS,'ATARI COMPUTER - MEMO PAD',CR
3505  F0F6  52 49 20 43
3506  F0FA  4F 4D 50 55
3507  F0FE  54 45 52 20
3508  F102  2D 20 4D 45
3509  F106  4D 4F 20 50
3510  F10A  41 44 9B
3511                        ;
3512  00F0                  IDENTH  =       IDENT/256
3513  00F2                  IDENTL  =       (-256)*IDENTH+IDENT ;SYSTEM I.D. MSG POINTER
3514                        ;
3515  000E                  TBLLEN  =       IDENT-TBLENT-1 ;HANDLER TABLE LENGTH
3516  F10D  42 4F 4F 54     DERR5:   .BYTE   'BOOT ERROR',CR
3517  F111  20 45 52 52
3518  F115  4F 52 9B
3519                        ;
3520  00F1                  DERRH   =       DERR5/256
3521  000D                  DERRL   =       (-256)*DERRH+DERR5 ;DISK ERROR MSG POINTER
3522                        ;
3523                        ;
3524                        ;
3525                        ;
3526                        ;       DEVICE/FILENAME SPECIFICATIONS
3527                        ;
3528  F11B  45 3A 9B        OPNEDT: .BYTE   'E:',CR     ;"OPEN SCREEN EDITOR" DEVICE SPEC.
3529                        ;
3530  00F1                  OPNH    =       OPNEDT/256
3531  0018                  OPNL    =(-256)*OPNH+OPNEDT ;SCREEN EDITOR OPEN POINTER
3532  F11B
3533                        ;
3534                        ;
3535                        ;
3536                        ;
3537                        ;*****************************************************************
3538                        ;       RESET BUTTON ROUTINE STARTS HERE
3539                        ;*****************************************************************
3540                        ;
3541  F11B  78              RESET:  SEI                 ;DISABLE IRQ INTERRUPTS
```

```
3542  F11C  AD 44 02          LDA     COLDST      ;WERE WE IN MIDDLE OF COLDSTART?
3543  F11F  D0 04             BNE     PWRUP       ;YES, GO TRY IT AGAIN
3544  F121  A9 FF             LDA     #$FF
3545  F123  D0 03             BNE     PWRUP1      ;SET WARM START FLAG
3546                        ;
3547                        ;
3548                        ;
3549                        ;*****************************************************************
3550                        ;       POWER UP ROUTINES START HERE
3551                        ;*****************************************************************
3552                        ;
3553  F125  78              PWRUP:  SEI             ;DISABLE IRQ INTERRUPTS
3554  F126  A9 00                   LDA     #0      ;CLEAR WARMSTART FLAG
3555  F128  85 08           PWRUP1: STA     WARMST
3556  F12A  D8                      CLD             ;CLEAR DECIMAL FLAG.
3557  F12B  A2 FF                   LDX     #$FF
3558  F12D  9A                      TXS             ;SET STACK POINTER
3559  F12E  20 3F F2                JSR     SPECL   ;CARTRIDGE SPECIAL CASE?
3560  F131  20 81 F2                JSR     HARDI   ;DO HARDWARE INITIALIZATION
3561  F134  A5 08                   LDA     WARMST  ;IS IT WARMSTART?
3562  F136  D0 28                   BNE     ZOSRAM  ;YES, ONLY ZERO OS RAM
3563                        ;
3564  F138  A9 00           ZERORM: LDA     #0
3565  F13A  A0 08                   LDY     #WARMST
3566  F13C  85 04                   STA     RAMLO
3567  F13E  85 05                   STA     RAMLO+1 ;INITIALIZE RAM POINTER
3568  F140  91 04           CLRRAM: STA     (RAMLO),Y ;CLEAR MEMORY LOC.
3569  F142  C8                      INY
3570  F143  C0 00                   CPY     #0      ;AT END OF PAGE?
3571  F145  D0 F9                   BNE     CLRRAM
3572  F147  E6 05                   INC     RAMLO+1 ;YES, INCR PAGE POINTER
3573  F149  A6 05                   LDX     RAMLO+1
3574  F14B  E4 06                   CPX     TRAMSZ  ;AT END OF MEM?
3575  F14D  D0 F1                   BNE     CLRRAM  ;NO.
3576                        ;
3577                        ; INITIALIZE DOSVEC TO POINT TO SIGNON (BLACKBOARD)
3578  F14F  AD 72 E4                LDA     BLKBDV+1
3579  F152  85 0A                   STA     DOSVEC  ;USE BLACKBOARD VECTOR
3580  F154  AD 73 E4                LDA     BLKBDV+2 ;FOR DOSVEC
3581  F157  85 0B                   STA     DOSVEC+1
3582  F159  A9 FF                   LDA     #$FF
3583  F15B  8D 44 02                STA     COLDST  ;SET TO SHOW IN MIDDLE OF COLDSTART
3584  F15E  D0 13                   BNE     ESTSCM  ;GO AROUND ZOSRAM
3585                        ;
3586                        ; CLEAR OS RAM (FOR WARMSTART)
3587  F160  A2 00           ZOSRAM: LDX     #0
3588  F162  8A                      TXA
3589  F163  9D 00 02        ZOSRM2: STA     $200,X  ;CLEAR PAGES 2 AND 3
3590  F166  9D 00 03                STA     $300,X
3591  F169  CA                      DEX
3592  F16A  D0 F7                   BNE     ZOSRM2
3593  F16C  A2 10                   LDX     #INTZBS
3594  F16E  95 00           ZOSRM3: STA     0,X     ;CLEAR ZERO PAGE LOCATIONS INTZBS-7F
3595  F170  E8                      INX
```

```
3596 F171 10 FB              BPL     ZOSRM3
3597                      ;
3598                      ; ESTABLISH SCREEN MARGINS
3599 F173 A9 02           ESTSCM: LDA     #LEDGE
3600 F175 85 52                   STA     LMARGN
3601 F177 A9 27                   LDA     #REDGE
3602 F179 85 53                   STA     RMARGN
3603                      ;
3604                      ;
3605                      ; MOVE VECTOR TABLE FROM ROM TO RAM
3606 F17B A2 25           OPSYS:  LDX     #$25
3607 F17D BD 80 E4        MOVVEC: LDA     VCTABL,X        ;ROM TABLE
3608 F180 9D 00 02                STA     INTABS,X        ;TO RAM
3609 F183 CA                      DEX
3610 F184 10 F7                   BPL     MOVVEC
3611 F186 20 94 F2                JSR     OSRAM           ;DO O.S. RAM SETUP
3612 F189 58                      CLI                     ;ENABLE IRQ INTERRUPTS
3613                      ;
3614                      ;
3615                      ;       LINK HANDLERS
3616                      ;
3617 F18A A2 0E                   LDX     #TBLLEN
3618 F18C BD E3 F0        NXTENT: LDA     TBLENT,X        ;READ HANDLER TABLE ENTRY
3619 F18F 9D 1A 03                STA     HATABS,X        ;PUT IN TABLE
3620 F192 CA                      DEX
3621 F193 10 F7                   BPL     NXTENT          ;DONE WITH ALL ENTRIES?
3622                      ;
3623                      ;
3624                      ;
3625                      ;
3626                      ;
3627                      ;       INTERROGATE CARTRIDGE ADDR. SPACE TO SEE WHICH CARTRIDGES THERE
3628                      ;
3629 F195 A2 00                   LDX     #0
3630 F197 86 07                   STX     TSTDAT          ;CLEAR "B" CART. FLAG
3631 F199 86 06                   STX     TRAMSZ          ;CLEAR "A" CART. FLAG
3632 F19B AE E4 02                LDX     RAMSIZ
3633 F19E E0 90                   CPX     #$90            ;RAM IN "B" CART. SLOT?
3634 F1A0 B0 0A                   BCS     ENDBCK
3635 F1A2 AD FC 9F                LDA     CART-$2000      ;NO,
3636 F1A5 D0 05                   BNE     ENDBCK          ;CART. PLUGGED INTO "B" SLOT?
3637 F1A7 E6 07                   INC     TSTDAT          ;YES, SET "B" CART. FLAG
3638 F1A9 20 3C F2                JSR     CBINI           ;INITIALIZE CARTRIDGE "B"
3639                      ;
3640 F1AC AE E4 02        ENDBCK: LDX     RAMSIZ
3641 F1AF E0 B0                   CPX     #$B0            ;RAM IN "A" CART. SLOT?
3642 F1B1 B0 0A                   BCS     ENDACK
3643 F1B3 AE FC BF                LDX     CART            ;NO,
3644 F1B6 D0 05                   BNE     ENDACK          ;CART. PLUGGED INTO "A" SLOT?
3645 F1B8 E6 06                   INC     TRAMSZ          ;YES, SET "A" CART. FLAG
3646 F1BA 20 39 F2                JSR     CAINI           ;INITIALIZE CARTRIDGE "A"
3647                      ;
3648                      ;
3649                      ;       OPEN SCREEN EDITOR
```

```
3650                                ;
3651  F1BD  A9 03          ENDACK: LDA     #3
3652  F1BF  A2 00                  LDX     #SEX
3653  F1C1  9D 42 03               STA     ICCOM,X     ;OPEN I/O COMMAND
3654  F1C4  A9 18                  LDA     #OPNL
3655  F1C6  9D 44 03               STA     ICBAL,X
3656  F1C9  A9 F1                  LDA     #OPNH
3657  F1CB  9D 45 03               STA     ICBAH,X     ;SET BUFFER POINTER TO OPEN SCREEN EDITOR
3658  F1CE  A9 0C                  LDA     #$C
3659  F1D0  9D 4A 03               STA     ICAX1,X     ;SET UP OPEN FOR INPUT/OUTPUT
3660  F1D3  20 56 E4               JSR     CIOV        ;GO TO CIO
3661                              ;
3662  F1D6  10 03                  BPL     SCRNOK      ;BR IF NO ERROR
3663  F1D8  4C 25 F1               JMP     PWRUP       ;RETRY PWRUP IF ERROR (SHOULD NEVER HAPPEN!)
3664  F1DB  E8              SCRNOK: INX                 ;SCREEN OK, SO WAIT FOR VBLANK TO
3665  F1DC  D0 FD                  BNE     SCRNOK      ;BRING UP THE DISPLAY
3666  F1DE  C8                     INY
3667  F1DF  10 FA                  BPL     SCRNOK
3668                              ;
3669                              ;
3670                              ; DO CASSETTE BOOT
3671  F1E1  20 B2 F3               JSR     CSBOOT      ;CHECK, BOOT, AND INIT
3672                              ;
3673                              ; CHECK TO SEE IF EITHER CARTRIDGE WANTS DISK BOOT
3674  F1E4  A5 06                  LDA     TRAMSZ      ;CHECK BOTH CARTRIDGES
3675  F1E6  05 07                  ORA     TSTDAT
3676  F1E8  F0 12                  BEQ     NOCART      ;NEITHER CARTRIDGE LIVES
3677  F1EA  A5 06                  LDA     TRAMSZ      ;"A" CART?
3678  F1EC  F0 03                  BEQ     NOA1        ;NO
3679  F1EE  AD FD BF               LDA     CARTFG      ;GET CARTRIDGE MODE FLAG
3680  F1F1  A6 07           NOA1:  LDX     TSTDAT      ;"B" CART?
3681  F1F3  F0 03                  BEQ     NOB1        ;NO
3682  F1F5  0D FD 9F               ORA     CARTFG-$2000;ADD OTHER FLAG
3683  F1F8  29 01           NOB1:  AND     #1          ;DOES EITHER CART WANT BOOT?
3684  F1FA  F0 03                  BEQ     NOBOOT      ;NO
3685                              ;
3686                              ; DO DISK BOOT
3687  F1FC  20 CF F2        NOCART: JSR    BOOT        ;CHECK, BOOT, AND INIT
3688                              ;
3689                              ; GO TO ONE OF THE CARTRIDGES IF THEY SO DESIRE
3690  F1FF  A9 00           NOBOOT: LDA    #0
3691  F201  8D 44 02               STA     COLDST      ;RESET TO SHOW DONE WITH COLDSTART
3692  F204  A5 06                  LDA     TRAMSZ      ;"A" CART?
3693  F206  F0 0A                  BEQ     NOA2        ;NO
3694  F208  AD FD BF               LDA     CARTFG      ;GET CARTRIDGE MODE FLAG
3695  F20B  29 04                  AND     #4          ;DOES IT WANT TO RUN?
3696  F20D  F0 03                  BEQ     NOA2        ;NO
3697  F20F  6C FA BF               JMP     (CARTCS)    ;RUN "A" CARTRIDGE
3698  F212  A5 07           NOA2:  LDA     TSTDAT      ;"B" CART?
3699  F214  F0 0A                  BEQ     NOCAR2      ;NO
3700  F216  AD FD 9F               LDA     CARTFG-$2000;GET "B" MODE FLAG
3701  F219  29 04                  AND     #4          ;DOES IT WANT TO RUN?
3702  F21B  F0 DF                  BEQ     NOCART      ;NO
3703  F21D  6C FA 9F               JMP     (CARTCS-$2000) ;RUN "B" CARTRIDGE
```

```
3704                            ;
3705                            ; NO CARTRIDGES, OR NEITHER WANTS TO RUN,
3706                            ; SO GO TO DOSVEC (DOS, CASSETTE, OR BLACKBOARD)
3707  F220  6C 0A 00           NOCAR2: JMP     (DOSVEC)
3708                            ;
3709                            ; PRINT SIGN-ON MESSAGE
3710  F223  A2 F2               SIGNON: LDX     #IDENTL
3711  F225  A0 F0                       LDY     #IDENTH
3712  F227  20 B5 F3                    JSR     PUTLIN          ;GO PUT SIGN-ON MSG ON SCREEN
3713                            ;
3714                            ;
3715                            ;
3716                            ;       BLACKBOARD ROUTINE
3717  F22A  20 30 F2           BLACKB: JSR     BLKB2           ;"JSR EGETCH"
3718  F22D  4C 2A F2                    JMP     BLACKB          ;FOREVER
3719  F230  AD 05 E4           BLKB2:  LDA     EDITRV+5        ;HIGH BYTE
3720  F233  48                         PHA
3721  F234  AD 04 E4                    LDA     EDITRV+4        ;LOW BYTE
3722  F237  48                         PHA
3723  F238  60                         RTS                     ;SIMULATES "JMP (EDITRV)"
3724                            ;
3725                            ;
3726                            ; CARTRIDGE INITIALIZATION INDIRECT JUMPS
3727  F239  6C FE BF           CAINI:  JMP     (CARTAD)
3728  F23C  6C FE 9F           CBINI:  JMP     (CARTAD-$2000)
```

```
3729                                    .PAGE
3730                                ;
3731                                ;
3732                                ;
3733                                ;
3734                                ;
3735                                ;
3736                                ;               S U B R O U T I N E S
3737                                ;
3738                                ;
3739                                ;
3740                                ;
3741                                ;
3742                                ;
3743                                ;
3744                                ;
3745                                ;
3746                                ;
3747                                ;
3748                                ;
3749                                ;
3750                                ;
3751                                ;
3752                                ;
3753                                ;
3754                                ;
3755                                ; CHECK FOR HOW MUCH RAM & SPECIAL CARTRIDGE CASE.
3756                                ; IF SPECIAL CARTRIDGE CASE, DON'T GO BACK -- GO TO CART.
3757                                ;
3758  F23F  AD FC BF                SPECL:  LDA     CART            ;CHECK FOR RAM OR CART
3759  F242  D0 12                           BNE     ENSPEC          ;GO IF NOTHING OR MAYBE RAM
3760  F244  EE FC BF                        INC     CART            ;NOW DO RAM CHECK
3761  F247  AD FC BF                        LDA     CART            ;IS IT ROM?
3762  F24A  D0 0A                           BNE     ENSPEC          ;NO
3763  F24C  AD FD BF                        LDA     CARTFG          ;YES,
3764  F24F  29 80                           AND     #$80            ;MASK OFF SPECIAL BIT
3765  F251  F0 03                           BEQ     ENSPEC          ;BIT SET?
3766  F253  6C FE BF                        JMP     (CARTAD)        ;YES, GO RUN CARTRIDGE
3767                                ;
3768                                ;       CHECK FOR AMOUNT OF RAM
3769                                ;
3770                                ;
3771  F256  CE FC BF                ENSPEC: DEC     CART            ;RESTORE RAM IF NEEDED
3772  F259  A9 00                           LDA     #0
3773  F25B  85 05                           STA     RAMLO+1
3774  F25D  A9 10                           LDA     #$10
3775  F25F  85 06                           STA     TRAMSZ          ;SET RAM POINTER TO 4K.
3776  F261  A0 00                           LDY     #0
3777  F263  B1 05                   HOWMCH: LDA     (RAMLO+1),Y     ;READ RAM LOCATION
3778  F265  85 07                           STA     TSTDAT          ;SAVE DATA
3779  F267  49 FF                           EOR     #$FF            ;INVERT IT.
3780  F269  85 04                           STA     RAMLO           ;SAVE INVERTED DATA
3781  F26B  91 05                           STA     (RAMLO+1),Y     ;WRITE INVERTED DATA.
3782  F26D  B1 05                           LDA     (RAMLO+1),Y     ;READ RAM AGAIN
```

```
3783  F26F  C5 04          CMP     RAMLO       ;IS IT THE INVERTED DATA?
3784  F271  D0 0D          BNE     ENDRAM
3785  F273  A5 07          LDA     TSTDAT      ;YES,
3786  F275  91 05          STA     (RAMLO+1),Y ;RESTORE ORIGINAL RAM DATA
3787  F277  A5 06          LDA     TRAMSZ
3788  F279  18             CLC
3789  F27A  69 10          ADC     #$10
3790  F27C  85 06          STA     TRAMSZ      ;INCR. RAM POINTER BY 4K.
3791  F27E  D0 E3          BNE     HOWMCH      ;GO FIND HOW MUCH RAM.
3792  F280  60             ENDRAM: RTS
3793                       ;
3794                       ;
3795                       ;
3796                       ;
3797                       ;       HARDWARE INITIALIZATION
3798                       ;
3799                       ;
3800  F281  A9 00          HARDI:  LDA     #0
3801  F283  AA                     TAX
3802  F284  9D 00 D0       CLRCHP: STA     $D000,X
3803  F287  9D 00 D4               STA     $D400,X
3804  F28A  9D 00 D2               STA     $D200,X
3805  F28D  9D 00 D3               STA     $D300,X
3806  F290  E8                     INX
3807  F291  D0 F1                  BNE     CLRCHP
3808  F293  60                     RTS
3809                       ;
3810                       ;
3811                       ;       O.S. RAM SETUP
3812                       ;
3813  F294  C6 11          OSRAM:  DEC     BRKKEY      ;TURN OFF BREAK KEY FLAG
3814  F296  A5 06                  LDA     TRAMSZ      ;READ RAM SIZE IN TEMP. REG.
3815  F298  8D E4 02               STA     RAMSIZ      ;SAVE IT IN RAM SIZE.
3816  F29B  8D E6 02               STA     MEMTOP+1    ;INIT. MEMTOP ADDR HI BYTE
3817  F29E  A9 00                  LDA     #0
3818  F2A0  8D E5 02               STA     MEMTOP      ;INIT. MEMTOP ADDR LO BYTE
3819  F2A3  A9 00                  LDA     #INIMLL
3820  F2A5  8D E7 02               STA     MEMLO
3821  F2A8  A9 07                  LDA     #INIMLH
3822  F2AA  8D E8 02               STA     MEMLO+1     ;INITIALIZE MEMLO ADDR VECTOR
3823  F2AD  20 0C E4               JSR     EDITRV+$C   ;EDITOR INIT.
3824  F2B0  20 1C E4               JSR     SCRENV+$C   ;SCREEN INIT.
3825  F2B3  20 2C E4               JSR     KEYBDV+$C   ;KEYBOARD INIT.
3826  F2B6  20 3C E4               JSR     PRINTV+$C   ;PRINTER HANDLER INIT
3827  F2B9  20 4C E4               JSR     CASETV+$C   ;CASSETTE HANDLER INIT
3828  F2BC  20 6E E4               JSR     CIOINV      ;CIO INIT.
3829  F2BF  20 65 E4               JSR     SIOINV      ;SIO INIT.
3830  F2C2  20 6B E4               JSR     INTINV      ;INTERRUPT HANDLER INIT.
3831  F2C5  AD 1F D0               LDA     CONSOL
3832  F2C8  29 01                  AND     #$1
3833  F2CA  D0 02                  BNE     NOKEY       ;GAME START KEY DEPRESSED?
3834  F2CC  E6 4A                  INC     CKEY        ;YES, SET KEY FLAG.
3835  F2CE  60             NOKEY:  RTS
3836
```

```
3837                                ;
3838                                ;       DO BOOT OF DISK
3839                                ;
3840  F2CF  A5 08           BOOT:   LDA     WARMST
3841  F2D1  F0 0A                   BEQ     NOWARM          ;WARM START?
3842  F2D3  A5 09                   LDA     BOOT?           ;YES,
3843  F2D5  29 01                   AND     #1
3844  F2D7  F0 03                   BEQ     NOINIT          ;VALID BOOT?
3845  F2D9  20 7E F3                JSR     DINI            ;YES, RE-INIT. DOS SOFTWARE
3846  F2DC  60              NOINIT: RTS
3847  F2DD  A9 01           NOWARM: LDA     #1
3848  F2DF  8D 01 03                STA     DUNIT           ;ASSIGN DISK DRIVE NO.
3849  F2E2  A9 53                   LDA     #STATC
3850  F2E4  8D 02 03                STA     DCOMND          ;SET UP STATUS COMMAND
3851  F2E7  20 53 E4                JSR     DSKINV          ;GO DO DISK STATUS
3852  F2EA  10 01                   BPL     DOBOOT          ;IS STATUS FROM SIO GOOD?
3853  F2EC  60                      RTS                     ;NO, GO BACK WITH BAD BOOT STATUS
3854                                ;
3855  F2ED  A9 00           DOBOOT: LDA     #0
3856  F2EF  8D 0B 03                STA     DAUX2
3857  F2F2  A9 01                   LDA     #1
3858  F2F4  8D 0A 03                STA     DAUX1           ;SET SECTOR # TO 1.
3859  F2F7  A9 00                   LDA     #BUFFL
3860  F2F9  8D 04 03                STA     DBUFLO
3861  F2FC  A9 04                   LDA     #BUFFH
3862  F2FE  8D 05 03                STA     DBUFHI          ;SET UP BUFFER ADDR
3863  F301  20 9D F3        SECT1:  JSR     GETSEC          ;GET SECTOR
3864  F304  10 08                   BPL     ALLSEC          ;STATUS O.K. ?
3865  F306  20 81 F3        BADDSK: JSR     DSKRDE          ;NO, GO PRINT DISK READ ERROR
3866  F309  A5 4B                   LDA     CASSBT
3867  F30B  F0 E0                   BEQ     DOBOOT          ;CASSETTE BOOT?
3868  F30D  60                      RTS                     ;YES, QUIT
3869  F30E  A2 03           ALLSEC: LDX     #3
3870  F310  BD 00 04        RDBYTE: LDA     CASBUF+3,X      ;READ A BUFFER BYTE
3871  F313  9D 40 02                STA     DFLAGS,X        ;STORE IT
3872  F316  CA                      DEX
3873  F317  10 F7                   BPL     RDBYTE          ;DONE WITH 4 BYTE TRANSFER ?
3874  F319  AD 42 02                LDA     BOOTAD          ;YES,
3875  F31C  85 04                   STA     RAMLO
3876  F31E  AD 43 02                LDA     BOOTAD+1
3877  F321  85 05                   STA     RAMLO+1         ;PUT BOOT ADDR INTO Z. PAGE RAM
3878  F323  AD 04 04                LDA     CASBUF+7        ;@(CASBUF+3)+4
3879  F326  85 0C                   STA     DOSINI          ;ESTABLISH DOS INIT ADDRESS
3880  F328  AD 05 04                LDA     CASBUF+8        ;(CASBUF+3)+5
3881  F32B  85 0D                   STA     DOSINI+1
3882  F32D  A0 7F           MVBUFF: LDY     #$7F            ;YES, SET BYTE COUNT
3883  F32F  B9 00 04        MVNXB:  LDA     CASBUF+3,Y
3884  F332  91 04                   STA     (RAMLO),Y       ;MOVE A BYTE FROM SECTOR BUFFER TO BOOT ADDR
3885  F334  88                      DEY
3886  F335  10 F8                   BPL     MVNXB           ;DONE ?
3887  F337  18                      CLC                     ;YES,
3888  F338  A5 04                   LDA     RAMLO
3889  F33A  69 80                   ADC     #$80
3890  F33C  85 04                   STA     RAMLO
```

```
3891  F33E  A5 05          LDA     RAMLO+1
3892  F340  69 00          ADC     #0
3893  F342  85 05          STA     RAMLO+1         ;INCR BOOT LOADER BUFFER POINTER.
3894  F344  CE 41 02        DEC     DBSECT          ;DECR # OF SECTORS.
3895  F347  F0 11           BEQ     ENBOOT          ;MORE SECTORS ?
3896  F349  EE 0A 03        INC     DAUX1           ;YES, INCR SECTOR #
3897  F34C  20 9D F3  SECTX: JSR     GETSEC          ;GO GET SECTOR.
3898  F34F  10 DC           BPL     MVBUFF          ;STATUS O.K. ?
3899  F351  20 81 F3        JSR     DSKRDE          ;NO, GO PRINT DISK READ ERROR
3900  F354  A5 4B           LDA     CASSBT
3901  F356  D0 AE           BNE     BADDSK          ;IF CASSETTE, QUIT.
3902  F358  F0 F2           BEQ     SECTX           ;IF DISK, TRY SECTOR AGAIN.
3903  F35A  A5 4B  ENBOOT:  LDA     CASSBT
3904  F35C  F0 03           BEQ     XBOOT           ;CASSETTE BOOT ?
3905  F35E  20 9D F3        JSR     GETSEC          ;YES, GET EOF RECORD, BUT DON'T USE IT.
3906  F361  20 6C F3  XBOOT: JSR     BLOAD           ;GO EXECUTE BOOT LOADER
3907  F364  B0 A0           BCS     BADDSK          ;IF BAD BOOT, DO IT OVER AGAIN
3908  F366  20 7E F3        JSR     DINI            ;GO INIT. SOFTWARE
3909  F369  E6 09           INC     BOOT?           ;SHOW BOOT SUCCESS
3910  F36B  60              RTS
3911  F36C  18      BLOAD:  CLC
3912  F36D  AD 42 02        LDA     BOOTAD
3913  F370  69 06           ADC     #6
3914  F372  85 04           STA     RAMLO
3915  F374  AD 43 02        LDA     BOOTAD+1
3916  F377  69 00           ADC     #0
3917  F379  85 05           STA     RAMLO+1         ;PUT START ADDR OF BOOTLOADER INTO RAM
3918  F37B  6C 04 00        JMP     (RAMLO)
3919  F37E  6C 0C 00  DINI:  JMP     (DOSINI)
3920                  ;
3921                  ;
3922                  ;
3923                  ;
3924                  ; DISPLAY DISK READ ERROR MSG
3925                  ;
3926  F381  A2 0D     DSKRDE: LDX     #DERRL
3927  F383  A0 F1            LDY     #DERRH
3928                  ;
3929                  ;
3930                  ;
3931                  ;       PUT LINE ON SCREEN AT PRESENT CURSOR POSITION
3932                  ;
3933                  ;               X-REG -- LO BYTE, BEGIN ADDR OF LINE
3934                  ;               Y-REG -- HI BYTE, BEGIN ADDR OF LINE
3935                  ;
3936  F385  8A       PUTLIN: TXA
3937  F386  A2 00            LDX     #SEX
3938  F388  9D 44 03         STA     ICBAL,X
3939  F38B  98               TYA
3940  F38C  9D 45 03         STA     ICBAH,X         ;SET UP ADDR OF BEGIN OF LINE
3941  F38F  A9 09            LDA     #PUTTXT
3942  F391  9D 42 03         STA     ICCOM,X         ;"PUT TEXT RECORD" COMMAND
3943  F394  A9 FF            LDA     #$FF
3944  F396  9D 48 03         STA     ICBLL,X         ;SET BUFFER LENGTH
```

```
3945  F399  20 56 E4          JSR     CIOV        ;PUT LINE ON SCREEN
3946  F39C  60                RTS
3947                  ;
3948                  ;
3949                  ;
3950                  ;
3951                  ;       GET SECTOR FROM DISK 0
3952                  ;
3953  F39D  A5 4B     GETSEC: LDA     CASSBT
3954  F39F  F0 03             BEQ     DISKM       ;CASSETTE BOOT ?
3955  F3A1  4C 7A E4          JMP     RBLOKV      ;YES, GO TO READ BLOCK ROUTINE
3956  F3A4  A9 52     DISKM:  LDA     #READ
3957  F3A6  8D 02 03          STA     DCOMND      ;SET READ SECTOR COMMAND
3958  F3A9  A9 01             LDA     #1
3959  F3AB  8D 01 03          STA     DUNIT       ;SET DRIVE NO. TO DRIVE 0
3960  F3AE  20 53 E4          JSR     DSKINV      ;GET SECTOR
3961  F3B1  60                RTS
3962                  ;
3963                  ;
3964                  ;
3965                  ; DO CHECK FOR CASSETTE BOOT & IF SO, DO BOOT
3966                  ;
3967  F3B2  A5 08     CSBOOT: LDA     WARMST      ;WARMSTART?
3968  F3B4  F0 0A             BEQ     CSBOT2      ;NO
3969  F3B6  A5 09             LDA     BOOT?       ;GET BOOT FLAG
3970  F3B8  29 02             AND     #2          ;WAS CASSETTE BOOT SUCCESFULL?
3971  F3BA  F0 03             BEQ     NOCSB2      ;NO
3972  F3BC  20 E1 F3          JSR     CINI        ;YES, INIT CASSETTE SOFTWARE
3973  F3BF  60        NOCSB2: RTS
3974                  ;
3975  F3C0  A5 4A     CSBOT2: LDA     CKEY
3976  F3C2  F0 1C             BEQ     NOCSBT      ;"C" KEY FLAG SET ?
3977  F3C4  A9 80             LDA     #$80        ;YES,
3978  F3C6  85 3E             STA     FTYPE       ;SET LONG IRG TYPE
3979  F3C8  E6 4B             INC     CASSBT      ;SET CASSETTE BOOT FLAG
3980  F3CA  20 7D E4          JSR     CSOPIV      ;OPEN CASSETTE FOR INPUT
3981  F3CD  20 01 F3          JSR     SECT1       ;DO BOOT & INIT.
3982  F3D0  A9 00             LDA     #0
3983  F3D2  85 4B             STA     CASSBT      ;RESET CASSETTE BOOT FLAG
3984  F3D4  85 4A             STA     CKEY        ;CLEAR KEY FLAG
3985  F3D6  06 09             ASL     BOOT?       ;SHIFT BOOT FLAG (NOW=2 IF SUCCESS)
3986  F3D8  A5 0C             LDA     DOSINI
3987  F3DA  85 02             STA     CASINI      ;MOVE INIT ADDRESS FOR CASSETTE
3988  F3DC  A5 0D             LDA     DOSINI+1
3989  F3DE  85 03             STA     CASINI+1
3990  F3E0  60        NOCSBT: RTS
3991                  ;
3992  F3E1  6C 02 00  CINI:   JMP     (CASINI)    ;INIT CASSETTE
3993                  ;*****************************************************************
3994                  ;
3995                  ;
3996                  ; SPARE BYTE OR MODULE TOO LONG FLAG
3997                  ;
3998  F3E4            CRNTP7  =       *
```

```
    3999                        ;
    4000                                *=$14
    4001  0014  00                MONSPR: .BYTE   KBDORG-CRNTP7 ;^GMONITP TOO LONG
    4002                        ;
```

```
4003                            .PAGE
4004                            .TITLE  'DISPLAY HANDLER -- 10-30-78  --  DISPLC'
4005                    ;
4006                    ; HANDLER DEPENDENT EQUATES
4007                    ;
4008  007D              CLRCOD  =       $7D             ;CLEAR SCREEN ATASCI CODE
4009  009F              CNTL1   =       $9F             ;POKEY KEY CODE FOR ^1
4010                    ;
4011  0068              FRMADR  =       SAVADR
4012  0066              TOADR   =       MLTTMP
4013                    ;
```

```
4014                            .PAGE
4015                    ;
4016                    ;
4017                            *=EDITRV
4018                    ;
4019                    ; SCREEN EDITOR HANDLER ENTRY POINT
4020                    ;
4021  E400  FB F3       EDITOR: .WORD   EOPEN-1
4022  E402  33 F6               .WORD   RETUR1-1      ;(CLOSE)
4023  E404  3D F6               .WORD   EGETCH-1
4024  E406  A3 F6               .WORD   EOUTCH-1
4025  E408  33 F6               .WORD   RETUR1-1      ;(STATUS)
4026  E40A  3C F6               .WORD   NOFUNC-1      ;(SPECIAL)
4027  E40C  4C E4 F3            JMP     PWRONA
4028  E40F  38                  .BYTE   $38           ;ROM FILLER BYTE
4029                    ;
4030                            *=SCRENV
4031                    ;
4032                    ; DISPLAY HANDLER ENTRY POINT
4033                    ;
4034  E410  F5 F3       DISPLA: .WORD   DOPEN-1
4035  E412  33 F6               .WORD   RETUR1-1      ;(CLOSE)
4036  E414  92 F5               .WORD   GETCH-1
4037  E416  B6 F5               .WORD   OUTCH-1
4038  E418  33 F6               .WORD   RETUR1-1      ;(STATUS)
4039  E41A  FB FC               .WORD   DRAW-1        ;(SPECIAL)
4040  E41C  4C E4 F3            JMP     PWRONA
4041  E41F  F6                  .BYTE   $F6           ;ROM FILLER BYTE
4042                    ;
4043                            *=KEYBDV
4044                    ;
4045                    ;
4046                    ; KEYBOARD HANDLER ENTRY POINT
4047                    ;
4048  E420  33 F6       KBDHND: .WORD   RETUR1-1
4049  E422  33 F6               .WORD   RETUR1-1      ;(CLOSE)
4050  E424  E1 F6               .WORD   KGETCH-1
4051  E426  3C F6               .WORD   NOFUNC-1      ;(OUTCH)
4052  E428  33 F6               .WORD   RETUR1-1      ;(STATUS)
4053  E42A  3C F6               .WORD   NOFUNC-1      ;(SPECIAL)
4054  E42C  4C E4 F3            JMP     PWRONA
4055  E42F  00                  .BYTE   0             ;ROM FILLER BYTE
4056                    ;
4057                    ;
4058                    ; INTERRUPT VECTOR TABLE ENTRY
4059                            *=VCTABL-INTABS+VKEYBD
4060  E488  BE FF               .WORD   PIRQQ         ;KEYBOARD IRQ INTERRUPT VECTOR
```

```
4061
4062                            *=KBDORG
4063                    ;
4064  F3E4  A9 FF       PWRONA: LDA     #$FF
4065  F3E6  8D FC 02             STA     CH
4066  F3E9  AD E6 02             LDA     MEMTOP+1
4067  F3EC  29 F0                AND     #$F0        ;INSURE 4K PAGE BOUNDARY
4068  F3EE  85 6A                STA     RAMTOP
4069  F3F0  A9 40                LDA     #$40        ;DEFAULT TO UPPER CASE ALPHA AT PWRON
4070  F3F2  8D BE 02             STA     SHFLOK
4071  F3F5  60                   RTS                 ;POWER ON COMPLETED
```

```
4072                                    .PAGE
4073                                ;
4074                                ;
4075                                ; BEGIN DISPLAY HANDLER OPEN PROCESSING
4076                                ;
4077 F3F6 A5 2B          DOPEN:    LDA     ICAX2Z          ;GET AUX 2 BYTE
4078 F3F8 29 0F                    AND     #$F
4079 F3FA D0 08                    BNE     OPNCOM          ;IF MODE ZERO, CLEAR ICAX1Z
4080 F3FC A5 2A          EOPEN:    LDA     ICAX1Z          ;CLEAR "CLR INHIBIT" AND "MXD MODE" BITS
4081 F3FE 29 0F                    AND     #$F
4082 F400 85 2A                    STA     ICAX1Z
4083 F402 A9 00                    LDA     #0
4084 F404 85 57          OPNCOM:   STA     DINDEX
4085 F406 A9 E0                    LDA     #$E0            ;INITIALIZE GLOBAL VBLANK RAM
4086 F408 8D F4 02                 STA     CHBAS
4087 F40B A9 02                    LDA     #2
4088 F40D 8D F3 02                 STA     CHACT
4089 F410 8D 2F 02                 STA     SDMCTL          ;TURN OFF DMA NEXT VBLANK
4090 F413 A9 01                    LDA     #SUCCES
4091 F415 85 4C                    STA     DSTAT           ;CLEAR STATUS
4092 F417 A9 C0                    LDA     #$C0            ;DO IRGEN
4093 F419 05 10                    ORA     POKMSK
4094 F41B 85 10                    STA     POKMSK
4095 F41D 8D 0E D2                 STA     IRGEN
4096 F420 A9 00                    LDA     #0
4097 F422 8D 93 02                 STA     TINDEX          ;TEXT INDEX MUST ALWAYS BE 0
4098 F425 85 64                    STA     ADRESS
4099 F427 85 7B                    STA     SWPFLG
4100 F429 8D F0 02                 STA     CRSINH          ;TURN CURSOR ON AT OPEN
4101 F42C A0 0E                    LDY     #14             ;CLEAR TAB STOPS
4102 F42E A9 01                    LDA     #1              ;INIT TAB STOPS TO EVERY 8 CHARACTERS
4103 F430 99 A3 02       CLRTBS:   STA     TABMAP,Y
4104 F433 88                       DEY
4105 F434 10 FA                    BPL     CLRTBS
4106 F436 A2 04                    LDX     #4              ;LOAD COLOR REGISTERS
4107 F438 BD C1 FE       DOPEN8:   LDA     COLRTB,X
4108 F43B 9D C4 02                 STA     COLOR0,X
4109 F43E CA                       DEX
4110 F43F 10 F7                    BPL     DOPEN8
4111 F441 A4 6A                    LDY     RAMTOP          ;DO TXTMSC=$2C40 (IF MEMTOP=3000)
4112 F443 88                       DEY
4113 F444 8C 95 02                 STY     TXTMSC+1
4114 F447 A9 60                    LDA     #$60
4115 F449 8D 94 02                 STA     TXTMSC
4116 F44C A6 57                    LDX     DINDEX
4117 F44E BD 69 FE                 LDA     ANCONV,X        ;CONVERT IT TO ANTIC CODE
4118 F451 D0 04                    BNE     DOPENA          ;IF ZERO, IT IS ILLEGAL
4119 F453 A9 91          OPNERR:   LDA     #BADMOD         ;SET ERROR STATUS
4120 F455 85 4C                    STA     DSTAT
4121 F457 85 51          DOPENA:   STA     HOLD1
4122 F459 A5 6A                    LDA     RAMTOP          ;SET UP AN INDIRECT POINTER
4123 F45B 85 65                    STA     ADRESS+1
4124 F45D BC 45 FE                 LDY     ALOCAT,X        ;ALLOCATE N BLOCKS OF 40 BYTES
4125 F460 A9 28          DOPEN1:   LDA     #40
```

```
4126  F462 20 21 F9          JSR   DBSUB
4127  F465 88                DEY
4128  F466 D0 F8             BNE   DOPEN1
4129  F468 AD 6F 02          LDA   GPRIOR      ;CLEAR GTIA MODES
4130  F46B 29 3F             AND   #$3F
4131  F46D 85 67             STA   OPNTMP+1
4132  F46F A8                TAY
4133  F470 E0 08             CPX   #8          ;TEST IF 320X1
4134  F472 90 17             BCC   NOT8
4135  F474 8A                TXA               ;GET 2 LOW BITS
4136  F475 6A                ROR   A
4137  F476 6A                ROR   A
4138  F477 6A                ROR   A
4139  F478 29 C0             AND   #$C0        ;NOW 2 TOP BITS
4140  F47A 05 67             ORA   OPNTMP+1
4141  F47C A8                TAY
4142  F47D A9 10             LDA   #16         ;SUBTRACT 16 MORE FOR PAGE BOUNDARY
4143  F47F 20 21 F9          JSR   DBSUB
4144  F482 E0 0B             CPX   #11         ;TEST MODE 11
4145  F484 D0 05             BNE   NOT8        ;IF MODE = 11
4146  F486 A9 06             LDA   #6          ;PUT GTIA LUM VALUE INTO BACKGROUND REGISTER
4147  F488 8D C8 02          STA   COLOR4
4148  F48B 8C 6F 02  NOT8:   STY   GPRIOR      ;STORE NEW PRIORITY
4149  F48E A5 64             LDA   ADRESS      ;SAVE MEMORY SCAN COUNTER ADDRESS
4150  F490 85 58             STA   SAVMSC
4151  F492 A5 65             LDA   ADRESS+1
4152  F494 85 59             STA   SAVMSC+1
4153  F496 AD 0B D4  VBWAIT: LDA   VCOUNT      ;WAIT FOR NEXT VBLANK BEFORE MESSING
4154  F499 C9 7A             CMP   #$7A        ;WITH THE DISPLAY LIST
4155  F49B D0 F9             BNE   VBWAIT
4156  F49D 20 1F F9          JSR   DBDEC       ;START PUTTING DISPLAY LIST RIGHT UNDER RAM
4157  F4A0 BD 75 FE          LDA   PAGETB,X    ;TEST IF DISPLAY LIST WILL BE IN TROUBLE
4158  F4A3 F0 06             BEQ   NOMOD       ;OF CROSSING A 256 BYTE PAGE BOUNDARY
4159  F4A5 A9 FF             LDA   #$FF        ;IF SO, DROP DOWN A PAGE
4160  F4A7 85 64             STA   ADRESS
4161  F4A9 C6 65             DEC   ADRESS+1
4162  F4AB A5 64  NOMOD:     LDA   ADRESS      ;SAVE END OF DISPLAY LIST FOR LATER
4163  F4AD 85 68             STA   SAVADR
4164  F4AF A5 65             LDA   ADRESS+1
4165  F4B1 85 69             STA   SAVADR+1
4166  F4B3 20 13 F9          JSR   DBDDEC      ;(DOUBLE BYTE DOUBLE DECREMENT)
4167  F4B6 A9 41             LDA   #$41        ;(ANTIC) WAIT FOR VBLANK AND JMP TO TOP
4168  F4B8 20 17 F9          JSR   STORE
4169  F4BB 86 66             STX   OPNTMP
4170  F4BD A9 18             LDA   #24         ;INITIALIZE BOTSCR
4171  F4BF 8D BF 02          STA   BOTSCR
4172  F4C2 A5 57             LDA   DINDEX      ;DISALLOW MIXED MODE IF MODE GE. 9
4173  F4C4 C9 09             CMP   #9
4174  F4C6 B0 2D             BCS   NOTMXD
4175  F4C8 A5 2A             LDA   ICAX1Z      ;TEST MIXED MODE
4176  F4CA 29 10             AND   #$10
4177  F4CC F0 27             BEQ   NOTMXD
4178  F4CE A9 04             LDA   #4
4179  F4D0 8D BF 02          STA   BOTSCR
```

```
4180 F4D3 A2 02          LDX     #2          ;ADD 4 LINES OF TEXT AT BOTTOM OF SCREEN
4181 F4D5 A9 02 DOPEN2:  LDA     #2
4182 F4D7 20 17 F9       JSR     STORE
4183 F4DA CA             DEX
4184 F4DB 10 F8          BPL     DOPEN2
4185 F4DD A4 6A          LDY     RAMTOP      ;RELOAD MSC FOR TEXT
4186 F4DF 88             DEY
4187 F4E0 98             TYA
4188 F4E1 20 17 F9       JSR     STORE
4189 F4E4 A9 60          LDA     #$60
4190 F4E6 20 17 F9       JSR     STORE
4191 F4E9 A9 42          LDA     #$42
4192 F4EB 20 17 F9       JSR     STORE
4193 F4EE 18             CLC
4194 F4EF A9 0C          LDA     #MXDMDE-NUMDLE ;POINT X AT MIXED MODE TABLE
4195 F4F1 65 66          ADC     OPNTMP
4196 F4F3 85 66          STA     OPNTMP
4197 F4F5 A4 66 NOTMXD:  LDY     OPNTMP
4198 F4F7 BE 51 FE       LDX     NUMDLE,Y    ;GET NUMBER OF DISPLAY LIST ENTRIES
4199 F4FA A5 51 DOPEN3:  LDA     HOLD1       ;STORE N DLE'S
4200 F4FC 20 17 F9       JSR     STORE
4201 F4FF CA             DEX
4202 F500 D0 F8          BNE     DOPEN3
4203 F502 A5 57          LDA     DINDEX      ;DO THE MESSY 320X1 PROBLEM
4204 F504 C9 08          CMP     #8
4205 F506 90 1C          BCC     DOPEN5
4206 F508 A2 5D          LDX     #93         ;GET REMAINING NUMBER OF DLE'S
4207 F50A A5 6A          LDA     RAMTOP      ;RELOAD MEMORY SCAN COUNTER
4208 F50C 38             SEC
4209 F50D E9 10          SBC     #$10
4210 F50F 20 17 F9       JSR     STORE
4211 F512 A9 00          LDA     #0
4212 F514 20 17 F9       JSR     STORE
4213 F517 A9 4F          LDA     #$4F        ;(ANTIC) RELOAD MSC CODE
4214 F519 20 17 F9       JSR     STORE
4215 F51C A5 51 DOPEN4:  LDA     HOLD1       ;DO REMAINING DLE'S
4216 F51E 20 17 F9       JSR     STORE
4217 F521 CA             DEX
4218 F522 D0 F8          BNE     DOPEN4
4219 F524 A5 59 DOPEN5:  LDA     SAVMSC+1    ;POLISH OFF DISPLAY LIST
4220 F526 20 17 F9       JSR     STORE
4221 F529 A5 58          LDA     SAVMSC
4222 F52B 20 17 F9       JSR     STORE
4223 F52E A5 51          LDA     HOLD1
4224 F530 09 40          ORA     #$40
4225 F532 20 17 F9       JSR     STORE
4226 F535 A9 70          LDA     #$70        ;24 BLANK LINES
4227 F537 20 17 F9       JSR     STORE
4228 F53A A9 70          LDA     #$70
4229 F53C 20 17 F9       JSR     STORE
4230 F53F A5 64          LDA     ADRESS      ;SAVE DISPLAY LIST ADDRESS
4231 F541 8D 30 02       STA     SDLSTL
4232 F544 A5 65          LDA     ADRESS+1
4233 F546 8D 31 02       STA     SDLSTL+1
```

```
4234  F549  A9 70          LDA   #$70        ;ADD LAST BLANK LINE ENTRY
4235  F54B  20 17 F9       JSR   STORE       ;POSITION ADRESS=SDLSTL-1
4236  F54E  A5 64          LDA   ADRESS      ;STORE NEW MEMTOP
4237  F550  8D E5 02       STA   MEMTOP
4238  F553  A5 65          LDA   ADRESS+1
4239  F555  8D E6 02       STA   MEMTOP+1
4240  F558  A5 68          LDA   SAVADR
4241  F55A  85 64          STA   ADRESS
4242  F55C  A5 69          LDA   SAVADR+1
4243  F55E  85 65          STA   ADRESS+1
4244  F560  AD 31 02       LDA   SDLSTL+1
4245  F563  20 17 F9       JSR   STORE
4246  F566  AD 30 02       LDA   SDLSTL
4247  F569  20 17 F9       JSR   STORE
4248  F56C  A5 4C          LDA   DSTAT       ;IF ERROR OCURRED ON ALLOCATION, OPEN THE ED
4249  F56E  10 07          BPL   DOPEN9
4250  F570  48             PHA               ;SAVE STATUS
4251  F571  20 FC F3       JSR   EOPEN       ;OPEN THE EDITOR
4252  F574  68             PLA               ;RESTORE STATUS
4253  F575  A8             TAY               ;AND RETURN IT TO CIO
4254  F576  60             RTS
4255  F577  A5 2A          DOPEN9: LDA   ICAX1Z      ;TEST CLEAR INHIBIT BIT
4256  F579  29 20          AND   #$20
4257  F57B  D0 0B          BNE   DOPEN7
4258  F57D  20 B9 F7       JSR   CLRSCR      ;CLEAR SCREEN
4259  F580  8D 90 02       STA   TXTROW      ;AND HOME TEXT CURSOR (AC IS ZERO)
4260  F583  A5 52          LDA   LMARGN
4261  F585  8D 91 02       STA   TXTCOL
4262  F588  A9 22          DOPEN7: LDA   #$22        ;EVERYTHING ELSE IS SET UP
4263  F58A  0D 2F 02       ORA   SDMCTL      ;SO TURN ON DMACTL
4264  F58D  8D 2F 02       STA   SDMCTL
4265  F590  4C 21 F6       JMP   RETURM
4266                       ;
4267                       ;
4268  F593  20 96 FA       GETCH:  JSR   RANGE       ;GETCH DOES INCRSR, GETPLT DOESN'T
4269  F596  20 A2 F5       JSR   GETPLT
4270  F599  20 32 FB       JSR   INATAC      ;CONVERT INTERNAL CODE TO ATASCII
4271  F59C  20 D4 F9       JSR   INCRSB
4272  F59F  4C 34 F6       JMP   RETUR1
4273  F5A2  20 47 F9       GETPLT: JSR   CONVRT      ;CONVERT ROW/COLUMN TO ADRESS
4274  F5A5  B1 64          LDA   (ADRESS),Y
4275  F5A7  2D A0 02       AND   DMASK
4276  F5AA  46 6F          SHIFTD: LSR   SHFAMT      ;SHIFT DATA DOWN TO LOW BITS
4277  F5AC  B0 03          BCS   SHIFT1
4278  F5AE  4A             LSR   A
4279  F5AF  10 F9          BPL   SHIFTD      ;(UNCONDITIONAL)
4280  F5B1  8D FA 02       SHIFT1: STA   CHAR
4281  F5B4  C9 00          CMP   #0          ;RESTORE FLAGS ALSO
4282  F5B6  60             RTS
4283                       ;
4284                       ;
4285                       ;
4286  F5B7  8D FB 02       OUTCH:  STA   ATACHR
4287  F5BA  20 96 FA       JSR   RANGE
```

```
4288                                ;       JSR     OFFCRS
4289  F5BD  AD FB 02                OUTCHA: LDA     ATACHR          ;TEST FOR CLEAR SCREEN
4290  F5C0  C9 7D                           CMP     #CLRCOD
4291  F5C2  D0 06                           BNE     OUTCHE
4292  F5C4  20 B9 F7                        JSR     CLRSCR
4293  F5C7  4C 21 F6                        JMP     RETURM
4294  F5CA  AD FB 02                OUTCHE: LDA     ATACHR          ;TEST FOR CARRIAGE RETURN
4295  F5CD  C9 9B                           CMP     #CR
4296  F5CF  D0 06                           BNE     OUTCHB
4297  F5D1  20 30 FA                        JSR     DOCRWS          ;DO CR
4298  F5D4  4C 21 F6                        JMP     RETURM
4299  F5D7  20 E0 F5                OUTCHB: JSR     OUTPLT
4300  F5DA  20 D8 F9                        JSR     INCRSR
4301  F5DD  4C 21 F6                        JMP     RETURM
4302                                ;
4303                                ;
4304  F5E0  AD FF 02                OUTPLT: LDA     SSFLAG          ;*****LOOP HERE IF START/STOP FLAG IS NON-0
4305  F5E3  D0 FB                           BNE     OUTPLT
4306  F5E5  A2 02                           LDX     #2
4307  F5E7  B5 54                   CRLOOP: LDA     ROWCRS,X        ;SAVE CURSOR LOCATION FOR DRAW LINE TO DRAW
4308  F5E9  95 5A                           STA     OLDROW,X
4309  F5EB  CA                              DEX
4310  F5EC  10 F9                           BPL     CRLOOP
4311  F5EE  AD FB 02                        LDA     ATACHR          ;CONVERT ATASCII(ATACHR) TO INTERNAL(CHAR)
4312  F5F1  A8                              TAY                     ;SAVE ATACHR
4313  F5F2  2A                              ROL     A
4314  F5F3  2A                              ROL     A
4315  F5F4  2A                              ROL     A
4316  F5F5  2A                              ROL     A
4317  F5F6  29 03                           AND     #3
4318  F5F8  AA                              TAX                     ;X HAS INDEX INTO ATAINT
4319  F5F9  98                              TYA                     ;RESTORE ATACHR
4320  F5FA  29 9F                           AND     #$9F            ;STRIP OFF COLUMN ADDRESS
4321  F5FC  1D F6 FE                        ORA     ATAINT,X        ;OR IN NEW COLUMN ADDRESS
4322  F5FF  8D FA 02                OUTCH2: STA     CHAR
4323  F602  20 47 F9                        JSR     CONVRT
4324  F605  AD FA 02                        LDA     CHAR
4325  F608  46 6F                   SHIFTU: LSR     SHFAMT          ;SHIFT UP TO PROPER POSITION
4326  F60A  B0 04                           BCS     SHIFT2
4327  F60C  0A                              ASL     A
4328  F60D  4C 08 F6                        JMP     SHIFTU
4329  F610  2D A0 02                SHIFT2: AND     DMASK
4330  F613  85 50                           STA     TMPCHR          ;SAVE SHIFTED DATA
4331  F615  AD A0 02                        LDA     DMASK           ;INVERT MASK
4332  F618  49 FF                           EOR     #$FF
4333  F61A  31 64                           AND     (ADRESS),Y      ;MASK OFF OLD DATA
4334  F61C  05 50                           ORA     TMPCHR          ;OR IN NEW DATA
4335  F61E  91 64                           STA     (ADRESS),Y
4336  F620  60                              RTS
4337                                ;
4338                                ;
4339  F621  20 A2 F5                RETURM: JSR     GETPLT          ;DO CURSOR ON THE WAY OUT
4340  F624  85 5D                           STA     OLDCHR
4341  F626  A6 57                           LDX     DINDEX          ;GRAPHICS HAVE INVISIBLE CURSOR
```

```
4342  F628  D0 0A          BNE     RETUR1
4343  F62A  AE F0 02       LDX     CRSINH      ;TEST CURSOR INHIBIT
4344  F62D  D0 05          BNE     RETUR1
4345  F62F  49 80          EOR     #$80        ;TOGGLE MSB
4346  F631  20 FF F5       JSR     OUTCH2      ;DISPLAY IT
4347  F634  A4 4C       RETUR1: LDY     DSTAT       ;RETURN TO CIO WITH STATUS IN Y
4348  F636  A9 01          LDA     #SUCCES
4349  F638  85 4C          STA     DSTAT       ;SET STATUS= SUCCESSFUL COMPLETION
4350  F63A  AD FB 02       LDA     ATACHR      ;PUT ATACHR IN AC FOR RETURN TO CIO
4351  F63D  60           NOFUNC: RTS             ;(NON-EXISTENT FUNCTION RETURN POINT)
4352                     ;
4353                     ;
4354                     ;
4355                     ; END OF DISPLAY HANDLER
4356                     ;
```

```
4357                                    .PAGE
4358                                ;
4359                                ;
4360                                ;
4361                                ;
4362  F63E  20 B3 FC        EGETCH: JSR     SWAP
4363  F641  20 88 FA                JSR     ERANGE
4364  F644  A5 6B                   LDA     BUFCNT          ;ANYTHING IN THE BUFFER?
4365  F646  D0 34                   BNE     EGETC3          ;YES
4366  F648  A5 54                   LDA     ROWCRS          ;NO, SO SAVE BUFFER START ADDRESS
4367  F64A  85 6C                   STA     BUFSTR
4368  F64C  A5 55                   LDA     COLCRS
4369  F64E  85 6D                   STA     BUFSTR+1
4370  F650  20 E2 F6        EGETC1: JSR     KGETCH          ;LET'S FILL OUR BUFFER
4371  F653  84 4C                   STY     DSTAT           ;SAVE KEYBOARD STATUS
4372  F655  AD FB 02                LDA     ATACHR          ;TEST FOR CR
4373  F658  C9 9B                   CMP     #CR
4374  F65A  F0 12                   BEQ     EGETC2
4375  F65C  20 AD F6                JSR     DOSS            ;NO, SO PRINT IT
4376  F65F  20 B3 FC                JSR     SWAP            ;JSR DOSS DID SWAP SO SWAP BACK
4377  F662  A5 63                   LDA     LOGCOL          ;BEEP IF NEARING LOGICAL COL 120
4378  F664  C9 71                   CMP     #113
4379  F666  D0 03                   BNE     EGETC6
4380  F668  20 0A F9                JSR     BELL
4381  F66B  4C 50 F6        EGETC6: JMP     EGETC1
4382  F66E  20 E4 FA        EGETC2: JSR     OFFCRS          ;GET BUFFER COUNT
4383  F671  20 00 FC                JSR     DOBUFC
4384  F674  A5 6C                   LDA     BUFSTR          ;RETURN A CHARACTER
4385  F676  85 54                   STA     ROWCRS
4386  F678  A5 6D                   LDA     BUFSTR+1
4387  F67A  85 55                   STA     COLCRS
4388  F67C  A5 6B           EGETC3: LDA     BUFCNT
4389  F67E  F0 11                   BEQ     EGETC5
4390  F680  C6 6B           EGETC7  DEC     BUFCNT          ;AND RETURN TILL BUFCNT=0
4391  F682  F0 0D                   BEQ     EGETC5
4392  F684  A5 4C                   LDA     DSTAT           ;IF ERROR, LOOP ON EGETC7 UNTIL BUFFER IS EM
4393  F686  30 F8                   BMI     EGETC7
4394  F688  20 93 F5                JSR     GETCH
4395  F68B  8D FB 02                STA     ATACHR
4396  F68E  4C B3 FC                JMP     SWAP            ;AND RETURN WITHOUT TURNING CURSOR BACK ON
4397  F691  20 30 FA        EGETC5: JSR     DOCRWS          ;DO REAL CARRIAGE RETURN
4398  F694  A9 9B                   LDA     #CR             ;AND RETURN EOL
4399  F696  8D FB 02                STA     ATACHR
4400  F699  20 21 F6                JSR     RETURM          ;TURN ON CURSOR THEN SWAP
4401  F69C  84 4C                   STY     DSTAT           ;SAVE KEYBOARD STATUS
4402  F69E  4C B3 FC                JMP     SWAP            ;AND RETURN THROUGH RETUR1
4403                                ;
4404  F6A1  6C 64 00        JSRIND: JMP     (ADRESS)        ;JSR TO THIS CAUSES JSR INDIRECT
4405                                ;
4406  F6A4  8D FB 02        EOUTCH: STA     ATACHR          ;SAVE ATASCII VALUE
4407  F6A7  20 B3 FC                JSR     SWAP
4408  F6AA  20 88 FA                JSR     ERANGE
4409  F6AD  20 E4 FA        DOSS:   JSR     OFFCRS          ;TURN OFF CURSOR
4410  F6B0  20 8D FC                JSR     TSTCTL          ;TEST FOR CONTROL CHARACTERS (Z=1 IF CTL)
```

```
4411  F6B3  F0 09          BEQ     EOUTC5
4412  F6B5  0E A2 02 EOUTC6: ASL     ESCFLG      ;ESCFLG ONLY WORKS ONCE
4413  F6B8  20 CA F5       JSR     OUTCHE
4414  F6BB  4C B3 FC ERETN: JMP     SWAP        ;AND RETURN THROUGH RETUR1
4415  F6BE  AD FE 02 EOUTC5: LDA     DSPFLG      ;DO DSPFLG AND ESCFLG
4416  F6C1  0D A2 02       ORA     ESCFLG
4417  F6C4  D0 EF          BNE     EOUTC6      ;IF NON-0 DISPLAY RATHER THAN EXECUTE IT
4418  F6C6  0E A2 02       ASL     ESCFLG
4419  F6C9  E8             INX                 ;PROCESS CONTROL CHARACTERS
4420  F6CA  BD C6 FE       LDA     CNTRLS,X    ;GET DISPLACEMENT INTO ROUTINE
4421  F6CD  85 64          STA     ADRESS
4422  F6CF  BD C7 FE       LDA     CNTRLS+1,X  ;GET HIGH BYTE
4423  F6D2  85 65          STA     ADRESS+1
4424  F6D4  20 A1 F6       JSR     JSRIND      ;DO COMPUTED JSR
4425  F6D7  20 21 F6       JSR     RETURM      ;DO CURSOR
4426  F6DA  4C B3 FC       JMP     SWAP        ;ALL DONE SO RETURN THROUGH RETUR1
4427                     ;
4428                     ;
4429                     ;
4430                     ;
4431                     ; END SCREEN EDITOR.
4432                     ;
4433                     ;
4434                     ; BEGIN KEYBOARD HANDLER
4435                     ;
4436                     ;
4437                     ;
4438                     ;
4439  F6DD  A9 FF          KGETC2: LDA     #$FF
4440  F6DF  8D FC 02       STA     CH
4441  F6E2  A5 2A          KGETCH: LDA     ICAX1Z      ;TEST LSB OF AUX1 FOR SPECIAL EDITOR READ MO
4442  F6E4  4A             LSR     A
4443  F6E5  B0 62          BCS     GETOUT
4444  F6E7  A9 80          LDA     #BRKABT
4445  F6E9  A6 11          LDX     BRKKEY      ;TEST BREAK
4446  F6EB  F0 58          BEQ     K7          ;IF BREAK, PUT BRKABT IN DSTAT AND CR IN ATA
4447  F6ED  AD FC 02       LDA     CH
4448  F6F0  C9 FF          CMP     #$FF
4449  F6F2  F0 EE          BEQ     KGETCH
4450  F6F4  85 7C          STA     HOLDCH      ;SAVE CH FOR SHIFT LOCK PROC
4451  F6F6  A2 FF          LDX     #$FF        ;"CLEAR" CH
4452  F6F8  8E FC 02       STX     CH
4453  F6FB  20 D8 FC       JSR     CLICK       ;DO KEYBOARD AUDIO FEEDBACK (A IS OK)
4454  F6FE  AA             KGETC3: TAX         ;DO ASCCON
4455  F6FF  E0 C0          CPX     #$C0        ;TEST FOR CTL & SHIFT TOGETHER
4456  F701  90 02          BCC     ASCC01
4457  F703  A2 03          LDX     #3          ;BAD CODE
4458  F705  BD FE FE ASCC01: LDA     ATASCI,X
4459  F708  8D FB 02       STA     ATACHR      ;DONE
4460  F70B  C9 80          CMP     #$80        ;DO NULLS
4461  F70D  F0 CE          BEQ     KGETC2
4462  F70F  C9 81          CMP     #$81        ;CHECK ATARI KEY
4463  F711  D0 0B          BNE     KGETC1
4464  F713  AD B6 02       LDA     INVFLG
```

```
4465 F716 49 80          EOR     #$80
4466 F718 8D B6 02       STA     INVFLG
4467 F71B 4C DD F6       JMP     KGETC2      ;DONT RETURN A VALUE
4468 F71E C9 82  KGETC1: CMP     #$82        ;CAPS/LOWER
4469 F720 D0 07          BNE     K1
4470 F722 A9 00          LDA     #0          ;CLEAR SHFLOK
4471 F724 8D BE 02       STA     SHFLOK
4472 F727 F0 B4          BEQ     KGETC2
4473 F729 C9 83  K1:     CMP     #$83        ;SHIFT CAPS/LOWER
4474 F72B D0 07          BNE     K2
4475 F72D A9 40          LDA     #$40
4476 F72F 8D BE 02       STA     SHFLOK      ;SHIFT BIT
4477 F732 D0 A9          BNE     KGETC2
4478 F734 C9 84  K2:     CMP     #$84        ;CNTL CAPS/LOWER
4479 F736 D0 07          BNE     K3
4480 F738 A9 80          LDA     #$80        ;CNTL BIT
4481 F73A 8D BE 02       STA     SHFLOK
4482 F73D D0 9E          BNE     KGETC2
4483 F73F C9 85  K3:     CMP     #$85        ;DO EOF
4484 F741 D0 0A          BNE     K6
4485 F743 A9 88          LDA     #EOFERR
4486 F745 85 4C  K7:     STA     DSTAT
4487 F747 85 11          STA     BRKKEY      ;RESTORE BREAK
4488 F749 A9 9B  GETOUT: LDA     #CR         ;PUT CR IN ATACHR
4489 F74B D0 26          BNE     K8          ;(UNCONDITIONAL)
4490 F74D A5 7C  K6:     LDA     HOLDCH      ;PROCESS SHIFT LOCKS
4491 F74F C9 40          CMP     #$40        ;REGULAR SHIFT AND CONTROL TAKE PRECEDENCE
4492 F751 B0 15          BCS     K5          ;OVER LOCK
4493 F753 AD FB 02       LDA     ATACHR      ;TEST FOR ALPHA
4494 F756 C9 61          CMP     #$61        ;LOWER CASE A
4495 F758 90 0E          BCC     K5          ;NOT ALPHA IF LT
4496 F75A C9 7B          CMP     #$7B        ;LOWER CASE Z+1
4497 F75C B0 0A          BCS     K5          ;NOT ALPHA IF GE
4498 F75E AD BE 02       LDA     SHFLOK      ;DO SHIFT/CONTROL LOCK
4499 F761 F0 05          BEQ     K5          ;IF NO LOCK, DONT RE-DO IT
4500 F763 05 7C          ORA     HOLDCH
4501 F765 4C FE F6       JMP     KGETC3      ;DO RETRY
4502 F768 20 8D FC  K5:  JSR     TSTCTL      ;DONT INVERT MSB OF CONTROL CHARACTERS
4503 F76B F0 09          BEQ     K4
4504 F76D AD FB 02       LDA     ATACHR
4505 F770 4D B6 02       EOR     INVFLG
4506 F773 8D FB 02  K8:  STA     ATACHR
4507 F776 4C 34 F6  K4:  JMP     RETUR1      ;ALL DONE
4508                     ;
4509                     ;
```

```
4510                            .PAGE
4511                    ;
4512                    ;
4513                    ; CONTROL CHARACTER PROCESSORS
4514                    ;
4515  F779  A9 80       ESCAPE: LDA     #$80            ;SET ESCAPE FLAG
4516  F77B  8D A2 02            STA     ESCFLG
4517  F77E  60                  RTS
4518  F77F  C6 54       CRSRUP: DEC     ROWCRS
4519  F781  10 06               BPL     COMRET
4520  F783  AE BF 02            LDX     BOTSCR          ;WRAPAROUND
4521  F786  CA                  DEX
4522  F787  86 54       UPDNCM: STX     ROWCRS
4523  F789  4C 5C FC   COMRET: JMP     STRBEG          ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4524  F78C  E6 54       CRSRDN: INC     ROWCRS
4525  F78E  A5 54               LDA     ROWCRS
4526  F790  CD BF 02            CMP     BOTSCR
4527  F793  90 F4               BCC     COMRET
4528  F795  A2 00               LDX     #0
4529  F797  F0 EE               BEQ     UPDNCM          ;(UNCONDITIONAL)
4530  F799  C6 55       CRSRLF: DEC     COLCRS
4531  F79B  A5 55               LDA     COLCRS
4532  F79D  30 04               BMI     CRSRL1          ;(IF LMARGN=0, THIS ELIMINATES PROBLEM CASE)
4533  F79F  C5 52               CMP     LMARGN
4534  F7A1  B0 04               BCS     COMRE1
4535  F7A3  A5 53       CRSRL1: LDA     RMARGN
4536  F7A5  85 55       LFRTCM: STA     COLCRS
4537  F7A7  4C DD FB    COMRE1: JMP     DOLCOL          ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4538  F7AA  E6 55       CRSRRT: INC     COLCRS
4539  F7AC  A5 55               LDA     COLCRS
4540  F7AE  C5 53               CMP     RMARGN
4541  F7B0  90 F5               BCC     COMRE1
4542  F7B2  F0 F3               BEQ     COMRE1          ;(CAUSE BLE)
4543  F7B4  A5 52               LDA     LMARGN
4544  F7B6  4C A5 F7            JMP     LFRTCM          ;UNCONDITIONAL TO COMMON STORE
4545  F7B9  20 F3 FC    CLRSCR: JSR     PUTMSC
4546  F7BC  A0 00               LDY     #0
4547  F7BE  98                  TYA                     ;PUT 0 IN THE AC
4548  F7BF  91 64       CLRSC2: STA     (ADRESS),Y      ;(AC IS ZERO)
4549  F7C1  C8                  INY
4550  F7C2  D0 FB               BNE     CLRSC2
4551  F7C4  E6 65               INC     ADRESS+1
4552  F7C6  A6 65               LDX     ADRESS+1
4553  F7C8  E4 6A               CPX     RAMTOP
4554  F7CA  90 F3               BCC     CLRSC2
4555  F7CC  A9 FF               LDA     #$FF            ;CLEAN UP LOGICAL LINE BIT MAP
4556  F7CE  99 B2 02    CLRSC3: STA     LOGMAP,Y        ;(Y IS ZERO AFTER CLRSC2 LOOP)
4557  F7D1  C8                  INY
4558  F7D2  C0 04               CPY     #4
4559  F7D4  90 F8               BCC     CLRSC3
4560  F7D6  20 E4 FC    HOME:   JSR     COLCR           ;PLACE COLCRS AT LEFT EDGE
4561  F7D9  85 63               STA     LOGCOL
4562  F7DB  85 6D               STA     BUFSTR+1
4563  F7DD  A9 00               LDA     #0
```

```
4564 F7DF 85 54          STA   ROWCRS
4565 F7E1 85 56          STA   COLCRS+1
4566 F7E3 85 6C          STA   BUFSTR
4567 F7E5 60             RTS
4568                     ;
4569 F7E6 A5 63          BS:   LDA   LOGCOL      ;BACKSPACE
4570 F7E8 C5 52                CMP   LMARGN
4571 F7EA F0 21                BEQ   BS1
4572 F7EC A5 55          BSA:  LDA   COLCRS      ;LEFT EDGE?
4573 F7EE C5 52                CMP   LMARGN
4574 F7F0 D0 03                BNE   BS3         ;NO
4575 F7F2 20 73 FC             JSR   DELTIM      ;YES, SEE IF LINE SHOULD BE DELETED
4576 F7F5 20 99 F7       BS3:  JSR   CRSRLF
4577 F7F8 A5 55                LDA   COLCRS
4578 F7FA C5 53                CMP   RMARGN
4579 F7FC D0 07                BNE   BS2
4580 F7FE A5 54                LDA   ROWCRS
4581 F800 F0 03                BEQ   BS2
4582 F802 20 7F F7             JSR   CRSRUP
4583 F805 A9 20          BS2:  LDA   #$20        ;MAKE BACKSPACE DESTRUCTIVE
4584 F807 8D FB 02             STA   ATACHR
4585 F80A 20 E0 F5             JSR   OUTPLT
4586 F80D 4C DD FB       BS1:  JMP   DOLCOL      ;AND RETURN
4587 F810 20 AA F7       TAB:  JSR   CRSRRT      ;BEGIN SEARCH
4588 F813 A5 55                LDA   COLCRS      ;TEST FOR NEW LINE
4589 F815 C5 52                CMP   LMARGN
4590 F817 D0 0A                BNE   TAB1        ;NO
4591 F819 20 34 FA             JSR   DOCR        ;DO CARRIAGE RETURN
4592 F81C 20 20 FB             JSR   LOGGET      ;CHECK IF END OF LOGICAL LINE
4593 F81F 90 02                BCC   TAB1        ;NO, CONTINUE
4594 F821 B0 07                BCS   TAB2        ;(UNCONDITIONAL)
4595 F823 A5 63          TAB1: LDA   LOGCOL      ;CHECK FOR TAB STOP
4596 F825 20 25 FB             JSR   BITGET
4597 F828 90 E6                BCC   TAB         ;NO, SO KEEP LOOKING
4598 F82A 4C DD FB       TAB2: JMP   DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4599 F82D A5 63          SETTAB: LDA LOGCOL
4600 F82F 4C 06 FB             JMP   BITSET      ;SET BIT IN MAP AND RETURN
4601 F832 A5 63          CLRTAB: LDA LOGCOL
4602 F834 4C 12 FB             JMP   BITCLR      ;CLEAR " " " " "
4603 F837 20 9D FC       INSCHR: JSR PHACRS
4604 F83A 20 A2 F5             JSR   GETPLT      ;GET CHARACTER UNDER CURSOR
4605 F83D 85 7D                STA   INSDAT
4606 F83F A9 00                LDA   #0
4607 F841 8D BB 02             STA   SCRFLG
4608 F844 20 FF F5       INSCH4: JSR OUTCH2      ;STORE DATA
4609 F847 A5 63                LDA   LOGCOL      ;SAVE LOGCOL: IF AFTER INCRSA LOGCOL IS
4610 F849 48                   PHA               ;< THAN IT IS NOW, END LOOP
4611 F84A 20 DC F9             JSR   INCRSA      ;SPECIAL INCRSR ENTRY POINT
4612 F84D 68                   PLA
4613 F84E C5 63                CMP   LOGCOL
4614 F850 B0 0C                BCS   INSCH3      ;QUIT
4615 F852 A5 7D          INSCH1: LDA INSDAT      ;KEEP GOING
4616 F854 48                   PHA
4617 F855 20 A2 F5             JSR   GETPLT
```

```
4618 F858 85 7D          STA   INSDAT
4619 F85A 68             PLA
4620 F85B 4C 44 F8       JMP   INSCH4
4621 F85E 20 A8 FC INSCH3: JSR  PLACRS
4622 F861 CE BB 02 INSCH6: DEC  SCRFLG
4623 F864 30 04          BMI   INSCH5     ;IF SCROLL OCCURRED
4624 F866 C6 54          DEC   ROWCRS     ;MOVE CURSOR UP
4625 F868 D0 F7          BNE   INSCH6     ;(UNCOND) CONTINUE UNTIL SCRFLG IS MINUS
4626 F86A 4C DD FB INSCH5: JMP  DOLCOL    ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4627                     ;
4628                     ;
4629 F86D 20 9D FC DELCHR: JSR  PHACRS
4630 F870 20 47 F9 DELCH1: JSR  CONVRT    ;GET DATA TO THE RIGHT OF THE CURSOR
4631 F873 A5 64          LDA   ADRESS
4632 F875 85 68          STA   SAVADR     ;SAVE ADRESS TO KNOW WHERE TO PUT DATA
4633 F877 A5 65          LDA   ADRESS+1
4634 F879 85 69          STA   SAVADR+1
4635 F87B A5 63          LDA   LOGCOL
4636 F87D 48             PHA
4637 F87E 20 D4 F9       JSR   INCRSB     ;PUT CURSOR OVER NEXT CHARACTER
4638 F881 68             PLA
4639 F882 C5 63          CMP   LOGCOL     ;TEST NEW LOGCOL AGAINST OLD LOGCOL
4640 F884 B0 10          BCS   DELCH2     ;IF OLD.GE.NEW THEN QUIT
4641 F886 A5 54          LDA   ROWCRS     ;IS ROW OFF SCREEN?
4642 F888 CD BF 02       CMP   BOTSCR
4643 F88B B0 09          BCS   DELCH2     ;YES, SO QUIT
4644 F88D 20 A2 F5       JSR   GETPLT     ;GET DATA UNDER CURSOR
4645 F890 A0 00          LDY   #0
4646 F892 91 68          STA   (SAVADR),Y ;PUT IT IN PREVIOUS POSITION
4647 F894 F0 DA          BEQ   DELCH1     ;AND LOOP (UNCONDITIONAL)
4648 F896 A0 00 DELCH2: LDY   #0
4649 F898 98             TYA
4650 F899 91 68          STA   (SAVADR),Y ;CLEAR THE LAST POSITION
4651 F89B 20 68 FC       JSR   DELTIA     ;TRY TO DELETE A LINE
4652 F89E 20 A8 FC       JSR   PLACRS
4653 F8A1 4C DD FB       JMP   DOLCOL     ;AND RETURN
4654 F8A4 38 INSLIN: SEC                  ;NORMAL INSLIN PUTS "1" INTO BIT MAP
4655 F8A5 20 7B FB INSLIA JSR  EXTEND     ;ENTRY POINT FOR C=0
4656 F8A8 A5 52          LDA   LMARGN     ;DO CARRIAGE RETURN (NO LF)
4657 F8AA 85 55          STA   COLCRS
4658 F8AC 20 47 F9       JSR   CONVRT     ;GET ADDRESS
4659 F8AF A5 64          LDA   ADRESS     ;SET UP TO=40+FROM (FROM = CURSOR)
4660 F8B1 85 68          STA   FRMADR
4661 F8B3 18             CLC
4662 F8B4 69 28          ADC   #40
4663 F8B6 85 66          STA   TOADR
4664 F8B8 A5 65          LDA   ADRESS+1
4665 F8BA 85 69          STA   FRMADR+1
4666 F8BC 69 00          ADC   #0
4667 F8BE 85 67          STA   TOADR+1
4668 F8C0 A6 54          LDX   ROWCRS     ;SET UP LOOP COUNTER
4669 F8C2 E0 17          CPX   #23
4670 F8C4 F0 08          BEQ   INSLI2
4671 F8C6 20 4E FB INSLI1: JSR  MOVLIN
```

```
4672  F8C9  E8              INX
4673  F8CA  E0 17           CPX     #23
4674  F8CC  D0 F8           BNE     INSLI1
4675  F8CE  20 9B FB  INSLI2: JSR    CLRLIN      ;CLEAR CURRENT LINE
4676  F8D1  4C DD FB        JMP     DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4677  F8D4  20 DD FB  DELLIN: JSR    DOLCOL      ;GET BEGINNING OF LOG LINE (HOLD1)
4678  F8D7  A4 51     DELLIA: LDY    HOLD1       ;SQUEEZE BIT MAP
4679  F8D9  84 54           STY     ROWCRS      ;PUT CURSOR THERE
4680  F8DB  A4 54     DELLIB: LDY    ROWCRS
4681  F8DD  98        DELLI1: TYA
4682  F8DE  38              SEC
4683  F8DF  20 23 FB        JSR     LO2GET      ;GET NEXT BIT
4684  F8E2  08              PHP
4685  F8E3  98              TYA
4686  F8E4  18              CLC
4687  F8E5  69 78           ADC     #120
4688  F8E7  28              PLP
4689  F8E8  20 04 FB        JSR     BITPUT      ;WRITE IT OVER PRESENT BIT
4690  F8EB  C8              INY
4691  F8EC  C0 18           CPY     #24
4692  F8EE  D0 ED           BNE     DELLI1      ;LOOP
4693  F8F0  AD B4 02        LDA     LOGMAP+2    ;SET LSB
4694  F8F3  09 01           ORA     #1
4695  F8F5  8D B4 02        STA     LOGMAP+2
4696  F8F8  A5 52     DELLI2: LDA    LMARGN      ;DELETE LINE OF DATA USING PART OF SCROLL
4697  F8FA  85 55           STA     COLCRS      ;CR NO LF
4698  F8FC  20 47 F9        JSR     CONVRT
4699  F8FF  20 B7 FB        JSR     SCROL1
4700  F902  20 20 FB        JSR     LOGGET      ;TEST NEXT LINE FOR CONTINUATION
4701                  ;IS IT A NEW LOG LINE?
4702  F905  90 D4           BCC     DELLIB      ;NO SO DELETE ANOTHER
4703  F907  4C DD FB        JMP     DOLCOL      ;YES SO DOLCOL AND RETURN
4704  F90A  A0 20     BELL:   LDY    #$20
4705  F90C  20 D8 FC  BELL1:  JSR    CLICK
4706  F90F  88              DEY
4707  F910  10 FA           BPL     BELL1
4708  F912  60              RTS
```

```
4709                                        .PAGE
4710                                ;
4711                                ;
4712                                ; ROUTINES
4713                                ;
4714                                ;
4715                                ; DOUBLE BYTE DECREMENT OF INDIRECT POINTER
4716                                ;INCLUDING DB SUBTRACT AND DB DOUBLE DECREMENT
4717                                ;
4718 F913 A9 02                    DBDDEC: LDA     #2
4719 F915 D0 0A                            BNE     DBSUB           ;(UNCONDITIONAL)
4720                                ;
4721                                ; STORE DATA INDIRECT AND DECREMENT POINTER
4722                                ; (PLACED HERE TO SAVE JMP DBDEC AFTER STORE)
4723 F917 A4 4C                    STORE:  LDY     DSTAT           ;RETURN ON ERROR
4724 F919 30 2B                            BMI     STROK
4725 F91B A0 00                            LDY     #0
4726 F91D 91 64                    STORE1: STA     (ADRESS),Y
4727                                ;           JMP     DBDEC           DECREMENT AND RETURN
4728                                ;
4729 F91F A9 01                    DBDEC:  LDA     #1
4730 F921 8D 9E 02                 DBSUB:  STA     SUBTMP
4731 F924 A5 4C                            LDA     DSTAT           ;RETURN ON ERROR
4732 F926 30 1E                            BMI     STROK
4733 F928 A5 64                            LDA     ADRESS
4734 F92A 38                               SEC
4735 F92B ED 9E 02                         SBC     SUBTMP
4736 F92E 85 64                            STA     ADRESS
4737 F930 B0 02                            BCS     DBSUB1
4738 F932 C6 65                            DEC     ADRESS+1
4739 F934 A5 0F                    DBSUB1: LDA     APPMHI+1        ;MAKE SURE NOTHING EVER OVERWRITES APPMHI
4740 F936 C5 65                            CMP     ADRESS+1
4741 F938 90 0C                            BCC     STROK           ;OK
4742 F93A D0 06                            BNE     STRERR          ;ERROR
4743 F93C A5 0E                            LDA     APPMHI
4744 F93E C5 64                            CMP     ADRESS
4745 F940 90 04                            BCC     STROK
4746 F942 A9 93                    STRERR: LDA     #SCRMEM         ;SHOW MEM TOO SMALL FOR SCREEN ERROR
4747 F944 85 4C                            STA     DSTAT
4748 F946 60                       STROK:  RTS
4749                                ;
4750                                ;
4751                                ;
4752                                ; CONVERT ROW/COLUMN CURSOR INTO REAL ADDRESS (FROM SAVMSC ON UP)
4753                                ;
4754 F947 A5 54                    CONVRT: LDA     ROWCRS          ;SAVE CURSOR
4755 F949 48                               PHA
4756 F94A A5 55                            LDA     COLCRS
4757 F94C 48                               PHA
4758 F94D A5 56                            LDA     COLCRS+1
4759 F94F 48                               PHA
4760 F950 20 F3 FC                         JSR     PUTMSC
4761 F953 A5 54                            LDA     ROWCRS          ;PUT 10*ROWCRS INTO MLTTMP
4762 F955 85 66                            STA     MLTTMP
```

```
4763  F957  A9 00          LDA   #0
4764  F959  85 67          STA   MLTTMP+1
4765  F95B  A5 66          LDA   MLTTMP      ;QUICK X8
4766  F95D  0A             ASL   A
4767  F95E  26 67          ROL   MLTTMP+1
4768  F960  85 51          STA   HOLD1       ;  (SAVE 2X VALUE)
4769  F962  A4 67          LDY   MLTTMP+1    ;  ""
4770  F964  8C 9F 02       STY   HOLD2       ;  ""
4771  F967  0A             ASL   A
4772  F968  26 67          ROL   MLTTMP+1
4773  F96A  0A             ASL   A
4774  F96B  26 67          ROL   MLTTMP+1
4775  F96D  18             CLC               ;ADD IN 2X
4776  F96E  65 51          ADC   HOLD1
4777  F970  85 66          STA   MLTTMP
4778  F972  A5 67          LDA   MLTTMP+1
4779  F974  6D 9F 02       ADC   HOLD2
4780  F977  85 67          STA   MLTTMP+1
4781  F979  A6 57          LDX   DINDEX      ;NOW SHIFT MLTTMP LEFT DHLINE TIMES TO FINIS
4782  F97B  BC 81 FE       LDY   DHLINE,X    ;MULTIPLY
4783  F97E  88       CONVR1: DEY             ;LOOP N TIMES
4784  F97F  30 07          BMI   CONVR2
4785  F981  06 66          ASL   MLTTMP
4786  F983  26 67          ROL   MLTTMP+1
4787  F985  4C 7E F9       JMP   CONVR1
4788  F988  BC A5 FE CONVR2: LDY  DIV2TB,X    ;NOW DIVIDE HCRSR TO ACCOUNT FOR PARTIAL BYT
4789  F98B  A5 55          LDA   COLCRS
4790  F98D  A2 07          LDX   #7          ;* TRICKY *
4791  F98F  88       CONVR3: DEY
4792  F990  30 0A          BMI   CONVR4
4793  F992  CA             DEX
4794  F993  46 56          LSR   COLCRS+1
4795  F995  6A             ROR   A
4796  F996  6E A1 02       ROR   TMPLBT      ;SAVE LOW BITS FOR MASK
4797  F999  4C 8F F9       JMP   CONVR3
4798  F99C  C8       CONVR4: INY             ;SO Y IS ZERO UPON RETURN FROM THIS ROUTINE
4799  F99D  18             CLC
4800  F99E  65 66          ADC   MLTTMP      ;ADD SHIFTED COLCRS TO MLTTMP
4801  F9A0  85 66          STA   MLTTMP
4802  F9A2  90 02          BCC   CONVR5
4803  F9A4  E6 67          INC   MLTTMP+1
4804  F9A6  38       CONVR5: SEC             ;* TRICKY *
4805  F9A7  6E A1 02 CONVR6  ROR   TMPLBT     ;SLIDE A "1" UP AGAINST LOW BITS (CONTINUE T
4806  F9AA  18             CLC
4807  F9AB  CA             DEX               ;AND FINISH SHIFT SO LOW BITS ARE
4808  F9AC  10 F9          BPL   CONVR6      ;RIGHT JUSTIFIED.
4809  F9AE  AE A1 02       LDX   TMPLBT      ;TMPLBT IS NOW THE INDEX INTO DMASKTB
4810  F9B1  A5 66          LDA   MLTTMP      ;PREPARE FOR RETURN
4811  F9B3  18             CLC
4812  F9B4  65 64          ADC   ADRESS
4813  F9B6  85 64          STA   ADRESS
4814  F9B8  85 5E          STA   OLDADR      ;REMEMBER THIS ADDRESS FOR CURSOR
4815  F9BA  A5 67          LDA   MLTTMP+1
4816  F9BC  65 65          ADC   ADRESS+1
```

```
4817 F9BE 85 65          STA     ADRESS+1
4818 F9C0 85 5F          STA     OLDADR+1
4819 F9C2 BD B1 FE       LDA     DMASKT,X
4820 F9C5 8D A0 02       STA     DMASK
4821 F9C8 85 6F          STA     SHFAMT
4822 F9CA 68             PLA
4823 F9CB 85 56          STA     COLCRS+1
4824 F9CD 68             PLA
4825 F9CE 85 55          STA     COLCRS
4826 F9D0 68             PLA
4827 F9D1 85 54          STA     ROWCRS
4828 F9D3 60             RTS
4829                  ;
4830                  ;
4831                  ; INCREMENT CURSOR AND DETECT BOTH END OF LINE AND END OF SCREEN
4832                  ;
4833 F9D4 A9 00       INCRSB: LDA     #0              ;NON-EXTEND ENTRY POINT
4834 F9D6 F0 02               BEQ     INCRSC
4835 F9D8 A9 9B       INCRSR: LDA     #$9B            ;SPECIAL CASE ELIMINATOR
4836 F9DA 85 7D       INCRSC: STA     INSDAT
4837 F9DC E6 63       INCRSA: INC     LOGCOL          ;(INSCHR ENTRY POINT)
4838 F9DE E6 55               INC     COLCRS
4839 F9E0 D0 02               BNE     INCRS2          ;DO HIGH BYTE
4840 F9E2 E6 56               INC     COLCRS+1
4841 F9E4 A5 55       INCRS2: LDA     COLCRS          ;TEST END OF LINE
4842 F9E6 A6 57               LDX     DINDEX
4843 F9E8 DD 8D FE            CMP     COLUMN,X        ;TEST TABLED VALUE FOR ALL SCREEN MODES
4844 F9EB F0 0B               BEQ     INC2A           ;DO CR IF EQUAL
4845 F9ED E0 00               CPX     #0              ;MODE 0?
4846 F9EF D0 06               BNE     INCRS3          ;IF NOT, JUST RETURN
4847 F9F1 C5 53               CMP     RMARGN          ;TEST AGAINST RMARGN
4848 F9F3 F0 02               BEQ     INCRS3          ;EQUAL IS OK
4849 F9F5 B0 01               BCS     INC2A           ;IF GREATER THAN, DO CR
4850 F9F7 60          INCRS3: RTS
4851 F9F8 E0 08       INC2A:  CPX     #8              ;CHECK MODE
4852 F9FA 90 04               BCC     DOCR1           ;NOT 320X1 SO DO IT
4853 F9FC A5 56               LDA     COLCRS+1        ;TEST MSD
4854 F9FE F0 F7               BEQ     INCRS3          ;ONLY AT 64 SO DON'T DO IT
4855 FA00 A5 57       DOCR1:  LDA     DINDEX          ;DONT MESS WITH LOGMAP IF NO MODE ZERO
4856 FA02 D0 30               BNE     DOCR
4857 FA04 A5 63               LDA     LOGCOL          ;TEST LINE OVERRUN
4858 FA06 C9 51               CMP     #81
4859 FA08 90 0A               BCC     DOCR1B          ;IF LESS THAN 81 IT IS DEFINITELY NOT LINE 3
4860 FA0A A5 7D               LDA     INSDAT
4861 FA0C F0 26               BEQ     DOCR            ;ONLY DO LOG LINE OVERFLOW IF INSDAT <>0
4862 FA0E 20 30 FA            JSR     DOCRWS          ;LOG LINE OVERFLOW IS SPECIAL CASE
4863 FA11 4C 77 FA            JMP     INCRS1          ;RETURN
4864 FA14 20 34 FA    DOCR1B: JSR     DOCR            ;GET IT OVER WITH
4865 FA17 A5 54               LDA     ROWCRS
4866 FA19 18                  CLC                     ;TEST LOGICAL LINE BIT MAP
4867 FA1A 69 78               ADC     #120
4868 FA1C 20 25 FB            JSR     BITGET
4869 FA1F 90 08               BCC     DOCR1A          ;DONT EXTEND IF OVERRUN IS INTO MIDDLE OF LO
4870 FA21 A5 7D               LDA     INSDAT          ;DONT EXTEND IF INSDAT IS ZERO
```

```
4871  FA23  F0 04          BEQ     DOCR1A      ;(INSCHR SPECIAL CASE)
4872  FA25  18             CLC                 ;INSERT "0" INTO BIT MAP
4873  FA26  20 A5 FB       JSR     INSLIA
4874  FA29  4C DD FB DOCR1A: JMP    DOLCOL      ;CONVERT ROW AND COL TO LOGCOL AND RETURN
4875  FA2C  A9 00    NOSCRL: LDA     #0          ;DOCR WITHOUT SCROLL
4876  FA2E  F0 02          BEQ     NOSCR1      ;(UNCONDITIONAL)
4877  FA30  A9 9B    DOCRWS: LDA     #$9B        ;DOCR WITH SCROLLING (NORMAL MODE)
4878  FA32  85 7D    NOSCR1: STA     INSDAT
4879  FA34  20 E4 FC DOCR:   JSR     COLCR       ;PLACE COLCRS AT LEFT EDGE
4880  FA37  A9 00          LDA     #0
4881  FA39  85 56          STA     COLCRS+1
4882  FA3B  E6 54          INC     ROWCRS
4883  FA3D  A6 57    DOCR2:  LDX     DINDEX
4884  FA3F  A0 18          LDY     #24         ;SET UP SCROLL LOOP COUNTER
4885  FA41  24 7B          BIT     SWPFLG
4886  FA43  10 05          BPL     DOCR2A      ;BRANCH IF NORMAL
4887  FA45  A0 04          LDY     #4
4888  FA47  98             TYA
4889  FA48  D0 03          BNE     DOCR2B      ;(UNCONDITIONAL)
4890  FA4A  BD 99 FE DOCR2A: LDA    NOROWS,X    ;GET NO OF ROWS
4891  FA4D  C5 54    DOCR2B: CMP     ROWCRS
4892  FA4F  D0 26          BNE     INCRS1
4893  FA51  8C 9D 02       STY     HOLD3
4894  FA54  8A             TXA                 ;DONT SCROLL IF MODE <> 0
4895  FA55  D0 20          BNE     INCRS1
4896  FA57  A5 7D          LDA     INSDAT      ;OR IF INSDAT = 0
4897  FA59  F0 1C          BEQ     INCRS1
4898                       LDA     INSDAT      IF INSDAT <> $9B THEN ROLL IN A 0
4899  FA5B  C9 9B          CMP     #$9B        ;TO EXTEND BOTTOM LOGICAL LINE
4900  FA5D  38             SEC
4901  FA5E  F0 01          BEQ     DOCR4B
4902  FA60  18             CLC
4903  FA61  20 AC FB DOCR4B: JSR    SCROLL      ;LOOP BACK TO HERE IF >1 SCROLLS
4904  FA64  EE BB 02       INC     SCRFLG
4905  FA67  C6 5C          DEC     BUFSTR      ;ROWS MOVE UP SO BUFSTR SHOULD TOO
4906  FA69  CE 9D 02       DEC     HOLD3
4907  FA6C  AD B2 02       LDA     LOGMAP
4908  FA6F  38             SEC                 ;FOR PARTIAL LINES, ROLL IN A "1"
4909  FA70  10 EF          BPL     DOCR4B      ;AGAIN IF PARTIAL LOGICAL LINE
4910  FA72  AD 9D 02       LDA     HOLD3       ;PLACE CURSOR AT NEW LINE NEAR THE BOTTOM
4911  FA75  85 54          STA     ROWCRS
4912  FA77  4C DD FB INCRS1: JMP    DOLCOL      ;COLVERT ROW AND COL TO LOGCOL AND RETURN
4913                       ;
4914                       ;
4915                       ; SUBEND: SUBTRACT ENDPT FROM ROWAC OR COLAC. (X=0 OR 2)
4916                       ;
4917  FA7A  38       SUBEND: SEC
4918  FA7B  B5 70          LDA     ROWAC,X
4919  FA7D  E5 74          SBC     ENDPT
4920  FA7F  95 70          STA     ROWAC,X
4921  FA81  B5 71          LDA     ROWAC+1,X
4922  FA83  E5 75          SBC     ENDPT+1
4923  FA85  95 71          STA     ROWAC+1,X
4924  FA87  60             RTS
```

```
4925                                ;
4926                                ;
4927                                ; RANGE: DO CURSOR RANGE TEST. IF ERROR, POP STACK TWICE AND JMP RETURN
4928                                ;(ERANGE IS EDITOR ENTRY POINT AND TEST IF EDITOR IS OPEN.
4929                                ;IF IT ISNT IT OPENS THE EDITOR AND CONTINUES)
4930                                ;
4931 FA88 AD BF 02                  ERANGE  LDA     BOTSCR          ;IF BOTSCR=4
4932 FA8B C9 04                             CMP     #4
4933 FA8D F0 07                             BEQ     RANGE           ;THEN IT IS IN MIXED MODE AND OK
4934 FA8F A5 57                             LDA     DINDEX          ;IF MODE = 0
4935 FA91 F0 03                             BEQ     RANGE           ;THEN IT IS IN EDITOR MODE AND OK
4936 FA93 20 FC F3                          JSR     EOPEN           ;IF NOT, OPEN EDITOR
4937 FA96 A9 27                  RANGE:     LDA     #39             ;***** RANGE CHECK RMARGN ***** SET UP AC
4938 FA98 C5 53                             CMP     RMARGN          ;***** RANGE CHECK RMARGN ***** COMPARE
4939 FA9A B0 02                             BCS     RANGE3          ;***** RANGE CHECK RMARGN ***** BRANCH GE
4940 FA9C 85 53                             STA     RMARGN          ;***** RANGE CHECK RMARGN ***** BAD SO STORE
4941 FA9E A6 57                  RANGE3:    LDX     DINDEX
4942 FAA0 BD 99 FE                          LDA     NOROWS,X        ;CHECK ROWS
4943 FAA3 C5 54                             CMP     ROWCRS
4944 FAA5 90 2A                             BCC     RNGERR          ;(ERROR IF TABLE.GE.ROWCRS)
4945 FAA7 F0 28                             BEQ     RNGERR
4946 FAA9 E0 08                             CPX     #8              ;CHECK FOR 320X1
4947 FAAB D0 0A                             BNE     RANGE1          ;SPECIAL CASE IT
4948 FAAD A5 56                             LDA     COLCRS+1
4949 FAAF F0 13                             BEQ     RNGOK           ;IF HIGH BYTE IS 0, COL IS OK
4950 FAB1 C9 01                             CMP     #1
4951 FAB3 D0 1C                             BNE     RNGERR          ;IF >1, BAD
4952 FAB5 F0 04                             BEQ     RANGE2          ;IF 1, GO CHECK LOW BYTE
4953 FAB7 A5 56                  RANGE1:    LDA     COLCRS+1        ;FOR OTHERS, NON-ZERO HIGH BYTE IS BAD
4954 FAB9 D0 16                             BNE     RNGERR
4955 FABB BD 8D FE               RANGE2:    LDA     COLUMN,X        ;CHECK LOW BYTE
4956 FABE C5 55                             CMP     COLCRS
4957 FAC0 90 0F                             BCC     RNGERR
4958 FAC2 F0 0D                             BEQ     RNGERR
4959 FAC4 A9 01                  RNGOK:     LDA     #SUCCES         ;SET STATUS OK
4960 FAC6 85 4C                             STA     DSTAT
4961 FAC8 A9 80                             LDA     #BRKABT         ;PREPARE BREAK ABORT STATUS
4962 FACA A6 11                             LDX     BRKKEY          ;CHECK BREAK KEY FLAG
4963 FACC 85 11                             STA     BRKKEY          ;'CLEAR' BREAK
4964 FACE F0 06                             BEQ     RNGER2          ;IF BREAK, QUIT IMMEDIATELY AND RETURN TO CI
4965 FAD0 60                                RTS
4966 FAD1 20 D6 F7               RNGERR:    JSR     HOME            ;ON RANGE ERROR, BRING CURSOR BACK
4967 FAD4 A9 8D                             LDA     #CRSROR         ;SHOW CURSOR OVERRANGE ERROR
4968 FAD6 85 4C                  RNGER2:    STA     DSTAT
4969 FAD8 68                     RNGER1:    PLA                     ;RESTORE STACK (THIS ROUTINE IS ALWAYS 1 LEV
4970 FAD9 68                                PLA                     ;AWAY FROM RETURN TO CIO)
4971 FADA A5 7B                             LDA     SWPFLG          ;IF SWAPPED, SWAP BACK
4972 FADC 10 03                             BPL     RETUR3
4973 FADE 20 B9 FC                          JSR     SWAPA           ;AND DONT DO RETUR1
4974 FAE1 4C 34 F6               RETUR3:    JMP     RETUR1          ;RETURN TO CIO
4975                                ;
4976                                ;
4977                                ;
4978                                ; OFFCRS: RESTORE OLD DATA UNDER CURSOR SO IT CAN BE MOVED
```

```
4979                            ;
4980  FAE4  A0 00               OFFCRS: LDY     #0
4981  FAE6  A5 5D                       LDA     OLDCHR
4982  FAE8  91 5E                       STA     (OLDADR),Y
4983  FAEA  60                          RTS
4984                            ;
4985                            ;
4986                            ;
4987                            ; BITMAP ROUTINES:
4988                            ;
4989                            ; BITCON: PUT MASK IN BITMSK AND INDEX IN X
4990                            ; BITPUT: PUT CARRY INTO BITMAP
4991                            ; BITROL: ROL CARRY INTO BOTTOM OF BITMAP (SCROLL)
4992                            ; BITSET: SET PROPER BIT
4993                            ; BITCLR: CLEAR PROPER BIT
4994                            ; BITGET: RETURN CARRY SET IF BIT IS THERE
4995                            ; LOGGET: DO BITGET FOR LOGMAP INSTEAD OF TABMAP
4996                            ;
4997  FAEB  48                  BITCON: PHA
4998  FAEC  29 07                       AND     #7
4999  FAEE  AA                          TAX                     ;GET MASK
5000  FAEF  BD B9 FE                    LDA     MASKTB,X
5001  FAF2  85 6E                       STA     BITMSK
5002  FAF4  68                          PLA                     ;PROCESS INDEX
5003  FAF5  4A                          LSR     A
5004  FAF6  4A                          LSR     A
5005  FAF7  4A                          LSR     A
5006  FAF8  AA                          TAX
5007  FAF9  60                          RTS
5008                            ;
5009                            ;
5010  FAFA  2E B4 02            BITROL: ROL     LOGMAP+2
5011  FAFD  2E B3 02                    ROL     LOGMAP+1
5012  FB00  2E B2 02                    ROL     LOGMAP
5013  FB03  60                          RTS
5014                            ;
5015                            ;
5016  FB04  90 0C               BITPUT: BCC     BITCLR          ;AND RETURN
5017                            ;OTHERWISE FALL THROUGH TO BITSET AND RETURN
5018  FB06  20 EB FA            BITSET: JSR     BITCON
5019  FB09  BD A3 02                    LDA     TABMAP,X
5020  FB0C  05 6E                       ORA     BITMSK
5021  FB0E  9D A3 02                    STA     TABMAP,X
5022  FB11  60                          RTS
5023                            ;
5024  FB12  20 EB FA            BITCLR: JSR     BITCON
5025  FB15  A5 6E                       LDA     BITMSK
5026  FB17  49 FF                       EOR     #$FF
5027  FB19  3D A3 02                    AND     TABMAP,X
5028  FB1C  9D A3 02                    STA     TABMAP,X
5029  FB1F  60                          RTS
5030                            ;
5031  FB20  A5 54               LOGGET: LDA     ROWCRS
5032  FB22  18                  LO1GET: CLC
```

```
5033  FB23  69 78          LO2GET: ADC     #120
5034  FB25  20 EB FA       BITGET: JSR     BITCON
5035  FB28  18                     CLC
5036  FB29  BD A3 02               LDA     TABMAP,X
5037  FB2C  25 6E                  AND     BITMSK
5038  FB2E  F0 01                  BEQ     BITGE1
5039  FB30  38                     SEC
5040  FB31  60             BITGE1: RTS
5041                       ;
5042                       ;
5043                       ;
5044                       ;
5045                       ; INATAC: INTERNAL(CHAR) TO ATASCII(ATACHR) CONVERSION
5046                       ;
5047  FB32  AD FA 02       INATAC: LDA     CHAR
5048  FB35  A4 57                  LDY     DINDEX      ;IF GRAPHICS MODES
5049  FB37  C0 03                  CPY     #3
5050  FB39  B0 0F                  BCS     INATA1      ;THEN DONT CHANGE CHAR
5051  FB3B  2A                     ROL     A
5052  FB3C  2A                     ROL     A
5053  FB3D  2A                     ROL     A
5054  FB3E  2A                     ROL     A
5055  FB3F  29 03                  AND     #3
5056  FB41  AA                     TAX
5057  FB42  AD FA 02               LDA     CHAR
5058  FB45  29 9F                  AND     #$9F
5059  FB47  1D FA FE               ORA     INTATA,X
5060  FB4A  8D FB 02       INATA1: STA     ATACHR
5061  FB4D  60                     RTS
5062                       ;
5063                       ;
5064                       ;
5065                       ; MOVLIN: MOVE 40 BYTES AT FRMADR TO TOADR SAVING OLD TOADR
5066                       ; DATA IN THE LINBUF.  THEN MAKE NEXT FRMADR
5067                       ; BE AT LINBUF FOR NEXT TRANSFER & TOADR=TOADR+40
5068                       ;
5069  FB4E  A9 02          MOVLIN  LDA     #LINBUF/256 ;SET UP ADRESS=LINBUF=$247
5070  FB50  85 65                  STA     ADRESS+1
5071  FB52  A9 47                  LDA     #LINBUF.AND.$FF
5072  FB54  85 64                  STA     ADRESS
5073  FB56  A0 27                  LDY     #39
5074  FB58  B1 66          MOVLI1: LDA     (TOADR),Y   ;SAVE TO DATA
5075  FB5A  85 50                  STA     TMPCHR
5076  FB5C  B1 68                  LDA     (FRMADR),Y  ;STORE DATA
5077  FB5E  91 66                  STA     (TOADR),Y
5078  FB60  A5 50                  LDA     TMPCHR
5079  FB62  91 64                  STA     (ADRESS),Y
5080  FB64  88                     DEY
5081  FB65  10 F1                  BPL     MOVLI1
5082  FB67  A5 65                  LDA     ADRESS+1    ;SET UP FRMADR=LAST LINE
5083  FB69  85 69                  STA     FRMADR+1
5084  FB6B  A5 64                  LDA     ADRESS
5085  FB6D  85 68                  STA     FRMADR
5086  FB6F  18                     CLC                 ;ADD 40 TO TOADR
```

```
5087  FB70  A5 66          LDA     TOADR
5088  FB72  69 28          ADC     #40
5089  FB74  85 66          STA     TOADR
5090  FB76  90 02          BCC     MOVLI2
5091  FB78  E6 67          INC     TOADR+1
5092  FB7A  60          MOVLI2: RTS
5093                      ;
5094                      ;
5095                      ;
5096                      ; EXTEND: EXTEND BIT MAP FROM ROWCRS (EXTEND LOGICAL LINE
5097                      ;
5098  FB7B  08          EXTEND: PHP                 ;SAVE CARRY
5099  FB7C  A0 17             LDY     #23
5100  FB7E  98          EXTEN1: TYA
5101  FB7F  20 22 FB          JSR     LOIGET
5102  FB82  08                PHP
5103  FB83  98                TYA
5104  FB84  18                CLC
5105  FB85  69 79             ADC     #121
5106  FB87  28                PLP
5107  FB88  20 04 FB          JSR     BITPUT
5108  FB8B  88          EXTEN3: DEY
5109  FB8C  30 04             BMI     EXTEN4
5110  FB8E  C4 54             CPY     ROWCRS
5111  FB90  B0 EC             BCS     EXTEN1
5112  FB92  A5 54       EXTEN4: LDA     ROWCRS
5113  FB94  18                CLC
5114  FB95  69 78             ADC     #120
5115  FB97  28                PLP
5116  FB98  4C 04 FB          JMP     BITPUT      ;STORE NEW LINE'S BIT AND RETURN
5117                      ;
5118                      ;
5119                      ;
5120                      ; CLRLIN: CLEAR LINE CURSOR IS ON
5121                      ;
5122  FB9B  A5 52       CLRLIN: LDA     LMARGN
5123  FB9D  85 55               STA     COLCRS
5124  FB9F  20 47 F9            JSR     CONVRT
5125  FBA2  A0 27               LDY     #39
5126  FBA4  A9 00               LDA     #0
5127  FBA6  91 64       CLRLI1: STA     (ADRESS),Y
5128  FBA8  88                  DEY
5129  FBA9  10 FB               BPL     CLRLI1
5130  FBAB  60                  RTS
5131                      ;
5132                      ;
5133                      ;
5134                      ;
5135                      ; SCROLL: SCROLL SCREEN
5136                      ;
5137  FBAC  20 FA FA      SCROLL: JSR     BITROL      ;ROLL IN CARRY
5138  FBAF  A5 58               LDA     SAVMSC      ;SET UP WORKING REGISTERS
5139  FBB1  85 64               STA     ADRESS
5140  FBB3  A5 59               LDA     SAVMSC+1
```

```
5141  FBB5  85 65          STA     ADRESS+1
5142  FBB7  A0 28  SCROL1: LDY     #40          ;LOOP
5143  FBB9  B1 64          LDA     (ADRESS),Y
5144  FBBB  A6 6A          LDX     RAMTOP       ;TEST FOR LAST LINE
5145  FBBD  CA             DEX
5146  FBBE  E4 65          CPX     ADRESS+1
5147  FBC0  D0 08          BNE     SCROL2
5148  FBC2  A2 D7          LDX     #$D7
5149  FBC4  E4 64          CPX     ADRESS
5150  FBC6  B0 02          BCS     SCROL2
5151  FBC8  A9 00          LDA     #0           ;YES SO STORE ZERO DATA FOR THIS ENTIRE LINE
5152  FBCA  A0 00  SCROL2: LDY     #0
5153  FBCC  91 64          STA     (ADRESS),Y
5154  FBCE  E6 64          INC     ADRESS
5155  FBD0  D0 E5          BNE     SCROL1
5156  FBD2  E6 65          INC     ADRESS+1
5157  FBD4  A5 65          LDA     ADRESS+1
5158  FBD6  C5 6A          CMP     RAMTOP
5159  FBD8  D0 DD          BNE     SCROL1
5160  FBDA  4C DD FB       JMP     DOLCOL       ;AND RETURN
5161                     ;
5162                     ;
5163                     ; DOLCOL: DO LOGICAL COLUMN FROM BITMAP AND COLCRS
5164                     ;
5165  FBDD  A9 00  DOLCOL: LDA     #0           ;START WITH ZERO
5166  FBDF  85 63          STA     LOGCOL
5167  FBE1  A5 54          LDA     ROWCRS
5168  FBE3  85 51          STA     HOLD1
5169  FBE5  A5 51  DOLCO1: LDA     HOLD1        ;ADD IN ROW COMPONENT
5170  FBE7  20 22 FB       JSR     LO1GET
5171  FBEA  B0 0C          BCS     DOLCO2       ;FOUND BEGINNING OF LINE
5172  FBEC  A5 63          LDA     LOGCOL       ;ADD 40 AND LOOK BACK ONE
5173  FBEE  18             CLC
5174  FBEF  69 28          ADC     #40
5175  FBF1  85 63          STA     LOGCOL
5176  FBF3  C6 51          DEC     HOLD1        ;UP ONE LINE
5177  FBF5  4C E5 FB       JMP     DOLCO1
5178  FBF8  18     DOLCO2: CLC                  ;ADD IN COLCRS
5179  FBF9  A5 63          LDA     LOGCOL
5180  FBFB  65 55          ADC     COLCRS
5181  FBFD  85 63          STA     LOGCOL
5182  FBFF  60             RTS
5183                     ;
5184                     ;
5185                     ;
5186                     ; DOBUFC: COMPUTE BUFFER COUNT AS THE NUMBER OF BYTES FROM
5187                     ; BUFSTR TO END OF LOGICAL LINE WITH TRAILING SPACES REMOVED
5188                     ;
5189  FC00  20 9D FC DOBUFC: JSR     PHACRS
5190  FC03  A5 63          LDA     LOGCOL
5191  FC05  48             PHA
5192  FC06  A5 6C          LDA     BUFSTR       ;START
5193  FC08  85 54          STA     ROWCRS
5194  FC0A  A5 6D          LDA     BUFSTR+1
```

```
5195  FC0C  85 55          STA     COLCRS
5196  FC0E  A9 01          LDA     #1
5197  FC10  85 6B          STA     BUFCNT
5198  FC12  A2 17  DOBUF1: LDX     #23          ;NORMAL
5199  FC14  A5 7B          LDA     SWPFLG       ;IF SWAPPED, ROW 3 IS THE LAST LINE ON SCREE
5200  FC16  10 02          BPL     DOB1
5201  FC18  A2 03          LDX     #3
5202  FC1A  E4 54  DOB1:   CPX     ROWCRS       ;TEST IF CRSR IS AT LAST SCREEN POSITION
5203  FC1C  D0 0B          BNE     DOBU1A
5204  FC1E  A5 55          LDA     COLCRS
5205  FC20  C5 53          CMP     RMARGN
5206  FC22  D0 05          BNE     DOBU1A
5207  FC24  E6 6B          INC     BUFCNT       ;YES SO FAKE INCRSR TO AVOID SCROLLING
5208  FC26  4C 39 FC       JMP     DOBUF2
5209  FC29  20 D4 F9 DOBU1A: JSR    INCRSB
5210  FC2C  E6 6B          INC     BUFCNT
5211  FC2E  A5 63          LDA     LOGCOL
5212  FC30  C5 52          CMP     LMARGN
5213  FC32  D0 DE          BNE     DOBUF1       ;NOT YET EOL
5214  FC34  C6 54          DEC     ROWCRS       ;BACK UP ONE INCRSR
5215  FC36  20 99 F7       JSR     CRSRLF
5216  FC39  20 A2 F5 DOBUF2: JSR    GETPLT       ;TEST CURRENT COLUMN FOR NON-ZERO DATA
5217  FC3C  D0 17          BNE     DOBUF4       ;QUIT IF NON-ZERO
5218  FC3E  C6 6B          DEC     BUFCNT       ;DECREMENT COUNTER
5219  FC40  A5 63          LDA     LOGCOL       ;BEGINNING OF LOGICAL LINE YET?
5220  FC42  C5 52          CMP     LMARGN
5221  FC44  F0 0F          BEQ     DOBUF4       ;YES, SO QUIT
5222  FC46  20 99 F7       JSR     CRSRLF       ;BACK UP CURSOR
5223  FC49  A5 55          LDA     COLCRS       ;IF LOGCOL=RMARGN, GO UP 1 ROW
5224  FC4B  C5 53          CMP     RMARGN
5225  FC4D  D0 02          BNE     DOBUF3
5226  FC4F  C6 54          DEC     ROWCRS
5227  FC51  A5 6B  DOBUF3: LDA     BUFCNT
5228  FC53  D0 E4          BNE     DOBUF2       ;LOOP UNLESS BUFCNT JUST WENT TO ZERO
5229  FC55  68     DOBUF4: PLA
5230  FC56  85 63          STA     LOGCOL
5231  FC58  20 A8 FC       JSR     PLACRS
5232  FC5B  60              RTS
5233                       ;
5234                       ;
5235                       ;
5236                       ;
5237                       ; STRBEG: MOVE BUFSTR TO BEGINNING OF LOGICAL LINE.
5238                       ;
5239  FC5C  20 DD FB STRBEG: JSR     DOLCOL       ;USE DOLCOL TO POINT HOLD1 AT BOL
5240  FC5F  A5 51          LDA     HOLD1
5241  FC61  85 6C          STA     BUFSTR
5242  FC63  A5 52          LDA     LMARGN
5243  FC65  85 6D          STA     BUFSTR+1
5244  FC67  60              RTS
5245                       ;
5246                       ;
5247                       ;
5248                       ;
```

```
5249                                ;
5250                                ; DELTIM: TIME TO DELETE A LINE IF IT IS EMPTY AND AN EXTENSION
5251                                ;
5252  FC68  A5 63           DELTIA: LDA     LOGCOL          ;IF LOGCOL<>LMARGN
5253  FC6A  C5 52                   CMP     LMARGN          ;THEN DONT MOVE UP ONE
5254  FC6C  D0 02                   BNE     DELTIB          ;LINE BEFORE TESTING DELTIM
5255  FC6E  C6 54                   DEC     ROWCRS
5256  FC70  20 DD F8         DELTIB: JSR     DOLCOL
5257  FC73  A5 63            DELTIM: LDA     LOGCOL          ;TEST FOR EXTENSION
5258  FC75  C5 52                   CMP     LMARGN
5259  FC77  F0 13                   BEQ     DELTI3          ;NO
5260  FC79  20 47 F9                JSR     CONVRT
5261  FC7C  A5 53                   LDA     RMARGN          ;SET UP COUNT
5262  FC7E  38                      SEC
5263  FC7F  E5 52                   SBC     LMARGN
5264  FC81  A8                      TAY
5265  FC82  B1 64            DELTI1: LDA     (ADRESS),Y
5266  FC84  D0 06                   BNE     DELTI3          ;FOUND A NON-0 SO QUIT AND RETURN
5267  FC86  88                      DEY
5268  FC87  10 F9                   BPL     DELTI1
5269  FC89  4C DB F8         DELTI2: JMP     DELLIB          ;DELETE A LINE AND RETURN
5270  FC8C  60               DELTI3: RTS
5271                                ;
5272                                ;
5273                                ;
5274                                ; TSTCTL: SEARCH CNTRLS TABLE TO SEE IF ATACHR IS A CNTL CHAR
5275                                ;
5276  FC8D  A2 2D            TSTCTL: LDX     #45             ;PREPARE TO SEARCH TABLE
5277  FC8F  BD C6 FE         TSTCT1: LDA     CNTRLS,X
5278  FC92  CD FB 02                 CMP     ATACHR
5279  FC95  F0 05                   BEQ     TSTCT2
5280  FC97  CA                      DEX
5281  FC98  CA                      DEX
5282  FC99  CA                      DEX
5283  FC9A  10 F3                   BPL     TSTCT1
5284  FC9C  60               TSTCT2: RTS
5285                                ;
5286                                ;
5287                                ;
5288                                ; PUSH ROWCRS,COLCRS AND COLCRS+1
5289                                ;
5290  FC9D  A2 02            PHACRS: LDX     #2
5291  FC9F  B5 54            PHACR1: LDA     ROWCRS,X
5292  FCA1  9D B8 02                 STA     TMPROW,X
5293  FCA4  CA                      DEX
5294  FCA5  10 F8                   BPL     PHACR1
5295  FCA7  60                      RTS
5296                                ;
5297                                ;
5298                                ; PULL COLCRS+1,COLCRS AND ROWCRS
5299                                ;
5300  FCA8  A2 02            PLACRS: LDX     #2
5301  FCAA  BD B8 02         PLACR1: LDA     TMPROW,X
5302  FCAD  95 54                   STA     ROWCRS,X
```

```
5303 FCAF CA                    DEX
5304 FCB0 10 F8                 BPL     PLACR1
5305 FCB2 60                    RTS
5306                    ;
5307                    ;
5308                    ;
5309                    ; SWAP: IF MIXED MODE, SWAP TEXT CURSORS WITH REGULAR CURSORS
5310                    ;
5311 FCB3 20 B9 FC     SWAP:   JSR     SWAPA       ;THIS ENTRY POINT DOES RETUR1
5312 FCB6 4C 34 F6              JMP     RETUR1
5313 FCB9 AD BF 02     SWAPA:  LDA     BOTSCR
5314 FCBC C9 18                 CMP     #24
5315 FCBE F0 17                 BEQ     SWAP3
5316 FCC0 A2 0B                 LDX     #11
5317 FCC2 B5 54         SWAP1:  LDA     ROWCRS,X
5318 FCC4 48                    PHA
5319 FCC5 BD 90 02              LDA     TXTROW,X
5320 FCC8 95 54                 STA     ROWCRS,X
5321 FCCA 68                    PLA
5322 FCCB 9D 90 02              STA     TXTROW,X
5323 FCCE CA                    DEX
5324 FCCF 10 F1                 BPL     SWAP1
5325 FCD1 A5 7B                 LDA     SWPFLG
5326 FCD3 49 FF                 EOR     #$FF
5327 FCD5 85 7B                 STA     SWPFLG
5328 FCD7 60            SWAP3:  RTS
5329                    ;
5330                    ;
5331                    ; CLICK: MAKE CLICK THROUGH KEYBOARD SPEAKER
5332                    ;
5333 FCD8 A2 7F         CLICK:  LDX     #$7F
5334 FCDA 8E 1F D0      CLICK1: STX     CONSOL
5335 FCDD 8E 0A D4              STX     WSYNC
5336 FCE0 CA                    DEX
5337 FCE1 10 F7                 BPL     CLICK1
5338 FCE3 60                    RTS
5339                    ;
5340                    ;
5341                    ; COLCR: PUTS EITHER 0 OR LMARGN INTO COLCRS BASED ON MODE AND SWPFLG
5342                    ;
5343 FCE4 A9 00         COLCR:  LDA     #0
5344 FCE6 A6 7B                 LDX     SWPFLG
5345 FCE8 D0 04                 BNE     COLCR1
5346 FCEA A6 57                 LDX     DINDEX
5347 FCEC D0 02                 BNE     COLCR2
5348 FCEE A5 52         COLCR1: LDA     LMARGN
5349 FCF0 85 55         COLCR2: STA     COLCRS
5350 FCF2 60                    RTS
5351                    ;
5352                    ;
5353                    ; PUTMSC: PUT SAVMSC INTO ADRESS
5354                    ;
5355 FCF3 A5 58         PUTMSC: LDA     SAVMSC      ;SET UP ADDRESS
5356 FCF5 85 64                 STA     ADRESS
```

```
5357  FCF7  A5 59          LDA     SAVMSC+1
5358  FCF9  85 65          STA     ADRESS+1
5359  FCFB  60             RTS
5360
```

```
5361                                        .PAGE
5362                                    ;
5363                                    ;
5364                                    ; DRAW -- DRAW A LINE FROM OLDROW,OLDCOL TO NEWROW,NEWCOL
5365                                    ; (THE AL MILLER METHOD FROM BASKETBALL)
5366  FCFC  A2 00                       DRAW:   LDX     #0
5367  FCFE  A5 22                               LDA     ICCOMZ          ;TEST COMMAND: $11=DRAW $12=FILL
5368  FD00  C9 11                               CMP     #$11
5369  FD02  F0 08                               BEQ     DRAWA
5370  FD04  C9 12                               CMP     #$12            ;TEST FILL
5371  FD06  F0 03                               BEQ     DRAWB           ;YES
5372  FD08  A0 84                               LDY     #NVALID         ;NO, SO RETURN INVALID COMMAND
5373  FD0A  60                                  RTS
5374  FD0B  E8                          DRAWB:  INX
5375  FD0C  8E B7 02                    DRAWA:  STX     FILFLG
5376  FD0F  A5 54                               LDA     ROWCRS          ;PUT CURSOR INTO NEWROW,NEWCOL
5377  FD11  85 60                               STA     NEWROW
5378  FD13  A5 55                               LDA     COLCRS
5379  FD15  85 61                               STA     NEWCOL
5380  FD17  A5 56                               LDA     COLCRS+1
5381  FD19  85 62                               STA     NEWCOL+1
5382  FD1B  A9 01                               LDA     #1
5383  FD1D  85 79                               STA     ROWINC          ;SET UP INITIAL DIRECTIONS
5384  FD1F  85 7A                               STA     COLINC
5385  FD21  38                                  SEC
5386  FD22  A5 60                               LDA     NEWROW          ;DETERMINE DELTA ROW
5387  FD24  E5 5A                               SBC     OLDROW
5388  FD26  85 76                               STA     DELTAR
5389  FD28  B0 0D                               BCS     DRAW1           ;DO DIRECTION AND ABSOLUTE VALUE
5390  FD2A  A9 FF                               LDA     #$FF            ;BORROW WAS ATTEMPTED
5391  FD2C  85 79                               STA     ROWINC          ;SET DIRECTION=DOWN
5392  FD2E  A5 76                               LDA     DELTAR
5393  FD30  49 FF                               EOR     #$FF            ;DELTAR = |DELTAR|
5394  FD32  18                                  CLC
5395  FD33  69 01                               ADC     #1
5396  FD35  85 76                               STA     DELTAR
5397  FD37  38                          DRAW1:  SEC
5398  FD38  A5 61                               LDA     NEWCOL          ;NOW DELTA COLUMN
5399  FD3A  E5 5B                               SBC     OLDCOL
5400  FD3C  85 77                               STA     DELTAC
5401  FD3E  A5 62                               LDA     NEWCOL+1        ;TWO-BYTE QUANTITY
5402  FD40  E5 5C                               SBC     OLDCOL+1
5403  FD42  85 78                               STA     DELTAC+1
5404  FD44  B0 16                               BCS     DRAW2           ;DIRECTION AND ABSOLUTE VALUE
5405  FD46  A9 FF                               LDA     #$FF            ;BORROW WAS ATTEMPTED
5406  FD48  85 7A                               STA     COLINC          ;SET DIRECTION = LEFT
5407  FD4A  A5 77                               LDA     DELTAC
5408  FD4C  49 FF                               EOR     #$FF            ;DELTAC = |DELTAC|
5409  FD4E  85 77                               STA     DELTAC
5410  FD50  A5 78                               LDA     DELTAC+1
5411  FD52  49 FF                               EOR     #$FF
5412  FD54  85 78                               STA     DELTAC+1
5413  FD56  E6 77                               INC     DELTAC          ;ADD ONE FOR TWOS COMPLEMENT
5414  FD58  D0 02                               BNE     DRAW2
```

```
5415  FD5A  E6 78          INC     DELTAC+1
5416  FD5C  A2 02    DRAW2: LDX     #2          ;ZERO RAM FOR DRAW LOOP
5417  FD5E  A0 00           LDY     #0
5418  FD60  84 73           STY     COLAC+1
5419  FD62  98       DRAW3A: TYA
5420  FD63  95 70           STA     ROWAC,X
5421  FD65  B5 5A           LDA     OLDROW,X
5422  FD67  95 54           STA     ROWCRS,X
5423  FD69  CA              DEX
5424  FD6A  10 F6           BPL     DRAW3A
5425  FD6C  A5 77           LDA     DELTAC      ;FIND LARGER ONE (ROW OR COL)
5426                        STA     COUNTR      (PREPARE COUNTR AND ENDPT)
5427                        STA     ENDPT
5428  FD6E  E8              INX                 ;MAKE X 0
5429  FD6F  A8              TAY
5430  FD70  A5 78           LDA     DELTAC+1
5431  FD72  85 7F           STA     COUNTR+1
5432  FD74  85 75           STA     ENDPT+1
5433  FD76  D0 0B           BNE     DRAW3       ;AUTOMATICALLY LARGER IF MSD>0
5434  FD78  A5 77           LDA     DELTAC
5435  FD7A  C5 76           CMP     DELTAR      ;LOW COL >LOW ROW?
5436  FD7C  B0 05           BCS     DRAW3       ;YES
5437  FD7E  A5 76           LDA     DELTAR
5438  FD80  A2 02           LDX     #2
5439  FD82  A8              TAY
5440  FD83  98       DRAW3: TYA                 ;PUT IN INITIAL CONDITIONS
5441  FD84  85 7E           STA     COUNTR
5442  FD86  85 74           STA     ENDPT
5443  FD88  48              PHA                 ;SAVE AC
5444  FD89  A5 75           LDA     ENDPT+1     ;PUT LSB OF HIGH BYTE
5445  FD8B  4A              LSR     A           ;INTO CARRY
5446  FD8C  68              PLA                 ;RESTORE AC
5447  FD8D  6A              ROR     A           ;ROR THE 9 BIT ACUMULATOR
5448  FD8E  95 70           STA     ROWAC,X
5449  FD90  A5 7E    DRAW4A: LDA    COUNTR      ;TEST ZERO
5450  FD92  05 7F           ORA     COUNTR+1
5451  FD94  D0 03           BNE     DRAW11      ;IF COUNTER IS ZERO, LEAVE DRAW
5452  FD96  4C 42 FE        JMP     DRAW10
5453  FD99  18       DRAW11: CLC                ;ADD ROW TO ROWAC (PLOT LOOP)
5454  FD9A  A5 70           LDA     ROWAC
5455  FD9C  65 76           ADC     DELTAR
5456  FD9E  85 70           STA     ROWAC
5457  FDA0  90 02           BCC     DRAW5
5458  FDA2  E6 71           INC     ROWAC+1
5459  FDA4  A5 71    DRAW5: LDA     ROWAC+1     ;COMPARE ROW TO ENDPOINT
5460  FDA6  C5 75           CMP     ENDPT+1     ;IF HIGH BYTE OF ROW IS .LT. HIGH
5461  FDA8  90 14           BCC     DRAW6       ;BYTE OF ENDPT, BLT TO COLUMN
5462  FDAA  D0 06           BNE     DRAW5A
5463  FDAC  A5 70           LDA     ROWAC
5464  FDAE  C5 74           CMP     ENDPT       ;LOW BYTE
5465  FDB0  90 0C           BCC     DRAW6       ;ALSO BLT
5466  FDB2  18       DRAW5A: CLC                ;GE SO MOVE POINT
5467  FDB3  A5 54           LDA     ROWCRS
5468  FDB5  65 79           ADC     ROWINC
```

```
5469 FDB7 85 54          STA   ROWCRS
5470 FDB9 A2 00          LDX   #0          ;AND SUBTRACT ENDPT FROM ROWAC
5471 FDBB 20 7A FA       JSR   SUBEND
5472 FDBE 18       DRAW6. CLC               ;DO SAME FOR COLUMN (DOUBLE BYTE ADD)
5473 FDBF A5 72          LDA   COLAC         ;ADD
5474 FDC1 65 77          ADC   DELTAC
5475 FDC3 85 72          STA   COLAC
5476 FDC5 A5 73          LDA   COLAC+1
5477 FDC7 65 78          ADC   DELTAC+1
5478 FDC9 85 73          STA   COLAC+1
5479 FDCB C5 75          CMP   ENDPT+1       ;COMPARE HIGH BYTE
5480 FDCD 90 27          BCC   DRAW8
5481 FDCF D0 06          BNE   DRAW6A
5482 FDD1 A5 72          LDA   COLAC         ;COMPARE LOW BYTE
5483 FDD3 C5 74          CMP   ENDPT
5484 FDD5 90 1F          BCC   DRAW8
5485 FDD7 24 7A    DRAW6A: BIT   COLINC      ;+ OR - ?
5486 FDD9 10 10          BPL   DRAW6B
5487 FDDB C6 55          DEC   COLCRS        ;DO DOUBLE BYTE DECREMENT
5488 FDDD A5 55          LDA   COLCRS
5489 FDDF C9 FF          CMP   #$FF
5490 FDE1 D0 0E          BNE   DRAW7
5491 FDE3 A5 56          LDA   COLCRS+1
5492 FDE5 F0 0A          BEQ   DRAW7         ;DONT DEC IF ZERO
5493 FDE7 C6 56          DEC   COLCRS+1
5494 FDE9 10 06          BPL   DRAW7         ;(UNCONDITIONAL)
5495 FDEB E6 55    DRAW6B: INC   COLCRS      ;DO DOUBLE BYTE INCREMENT
5496 FDED D0 02          BNE   DRAW7
5497 FDEF E6 56          INC   COLCRS+1
5498 FDF1 A2 02    DRAW7:  LDX   #2          ;AND SUBTRACT ENDPT FROM COLAC
5499 FDF3 20 7A FA       JSR   SUBEND
5500 FDF6 20 96 FA DRAW8:  JSR   RANGE
5501 FDF9 20 E0 F5       JSR   OUTPLT        ;PLOT POINT
5502 FDFC AD B7 02       LDA   FILFLG        ;TEST RIGHT FILL
5503 FDFF F0 2F          BEQ   DRAW9
5504 FE01 20 9D FC       JSR   PHACRS
5505 FE04 AD FB 02       LDA   ATACHR
5506 FE07 8D BC 02       STA   HOLD4
5507 FE0A A5 54    DRAW8A: LDA   ROWCRS      ;SAVE ROW IN CASE OF CR
5508 FE0C 48             PHA
5509 FE0D 20 DC F9       JSR   INCRSA        ;POSITION CURSOR ONE PAST DOT
5510 FE10 68             PLA                 ;RESTORE ROWCRS
5511 FE11 85 54          STA   ROWCRS
5512 FE13 20 96 FA DRAW8C: JSR   RANGE
5513 FE16 20 A2 F5       JSR   GETPLT        ;GET DATA
5514 FE19 D0 0C          BNE   DRAW8B        ;STOP IF NON-ZERO DATA IS ENCOUNTERED
5515 FE1B AD FD 02       LDA   FILDAT        ;FILL DATA
5516 FE1E 8D FB 02       STA   ATACHR
5517 FE21 20 E0 F5       JSR   OUTPLT        ;DRAW IT
5518 FE24 4C 0A FE       JMP   DRAW8A        ;LOOP
5519 FE27 AD BC 02 DRAW8B: LDA   HOLD4
5520 FE2A 8D FB 02       STA   ATACHR
5521 FE2D 20 A8 FC       JSR   PLACRS
5522 FE30 38       DRAW9:  SEC               ;DO DOUBLE BYTE SUBTRACT
```

```
5523  FE31  A5 7E          LDA   COUNTR
5524  FE33  E9 01          SBC   #1
5525  FE35  85 7E          STA   COUNTR
5526  FE37  A5 7F          LDA   COUNTR+1
5527  FE39  E9 00          SBC   #0
5528  FE3B  85 7F          STA   COUNTR+1
5529  FE3D  30 03          BMI   DRAW1O
5530  FE3F  4C 90 FD       JMP   DRAW4A
5531  FE42  4C 34 F6 DRAW1O: JMP  RETUR1
```

```
5532                                        PAGE
5533                                ;
5534                                ;
5535                                ; TABLES
5536                                ;
5537                                ;
5538                                ; MEMORY ALLOCATION
5539                                ;
5540 FE45 18 10 0A 0A       ALOCAT: .BYTE   24,16,10,10,16,28,52,100,196,196,196,196
5541 FE49 10 1C 34 64
5542 FE4D C4 C4 C4 C4
5543                                ;
5544                                ;
5545                                ; NUMBER OF DISPLAY LIST ENTRIES
5546                                ;
5547 FE51 17 17 0B 17       NUMDLE: .BYTE   23,23,11,23,47,47,95,95,97,97,97,97
5548 FE55 2F 2F 5F 5F
5549 FE59 61 61 61 61
5550 FE5D 13 13 09 13       MXDMDE: .BYTE   19,19,9,19,39,39,79,79,65,65,65,65 ;(EXTENSION OF NUMDLE
5551 FE61 27 27 4F 4F
5552 FE65 41 41 41 41
5553                                ;
5554                                ;
5555                                ; ANTIC CODE FROM INTERNAL MODE CONVERSION TABLE
5556                                ;
5557                                ;   INTERNAL          ANTIC CODE              DESCRIPTION
5558                                ;       0                 2                40X2X8  CHARACTERS
5559                                ;       1                 6                20X5X8      ""
5560                                ;       2                 7                20X5X16     ""
5561                                ;       3                 8                40X4X8  GRAPHICS
5562                                ;       4                 9                80X2X4      ""
5563                                ;       5                 A                80X4X4      ""
5564                                ;       6                 B                160X2X2     ""
5565                                ;       7                 D                160X4X2     ""
5566                                ;       8                 F                320X2X1     ""
5567                                ;       9                 SAME AS 8 BUT GTIA 'LUM' MODE
5568                                ;       10                SAME AS 8 BUT GTIA 'COL/LUM REGISTER' MODE
5569                                ;       11                SAME AS 8 BUT GTIA 'COLOR' MODE
5570                                ;
5571 FE69 02 06 07 08       ANCONV: .BYTE   2,6,7,8,9,$A,$B,$D,$F,$F,$F,$F ;ZEROS FOR RANGE TEST IN
5572 FE6D 09 0A 0B 0D
5573 FE71 0F 0F 0F 0F
5574                                ;
5575                                ;
5576                                ; PAGE TABLE TELLS WHICH DISPLAY LISTS ARE IN DANGER OF
5577                                ; CROSSING A 256 BYTE PAGE BOUNDARY
5578                                ;
5579 FE75 00 00 00 00       PAGETB: .BYTE   0,0,0,0,0,0,0,1,1,1,1,1
5580 FE79 00 00 00 01
5581 FE7D 01 01 01 01
5582                                ;
5583                                ;
5584                                ; THIS IS THE NUMBER OF LEFT SHIFTS NEEDED TO MULTIPLY
5585                                ; COLCRS BY 10,20, OR 40.   (ROWCRS*10)/(2**DHLINE)
```

```
5586                            ;
5587  FE81  02 01 01 00        DHLINE: .BYTE   2,1,1,0,0,1,1,2,2,2,2,2
5588  FE85  00 01 01 02
5589  FE89  02 02 02 02
5590                            ;
5591                            ;
5592                            ; COLUMN: NUMBER OF COLUMNS
5593                            ;
5594  FE8D  28 14 14 28        COLUMN: .BYTE   40,20,20,40,80,80,160,160,64,80,80,80 ;MODE 8 IS SPECIAL
5595  FE91  50 50 A0 A0
5596  FE95  40 50 50 50
5597                            ;
5598                            ;
5599                            ;
5600                            ; NOROWS: NUMBER OF ROWS
5601                            ;
5602  FE99  18 18 0C 18        NOROWS: .BYTE   24,24,12,24,48,48,96,96,192,192,192,192
5603  FE9D  30 30 60 60
5604  FEA1  C0 C0 C0 C0
5605                            ;
5606                            ;
5607                            ;
5608                            ;
5609                            ; DIV2TB: HOW MANY RIGHT SHIFTS FOR HCRSR FOR PARTIAL BYTE MODES
5610                            ;
5611  FEA5  00 00 00 02        DIV2TB: .BYTE   0,0,0,2,3,2,3,2,3,1,1,1
5612  FEA9  03 02 03 02
5613  FEAD  03 01 01 01
5614                            ;
5615                            ;
5616                            ; DMASKT: DISPLAY MASK TABLE
5617                            ;
5618  FEB1  00 FF F0 0F        DMASKT: .BYTE   $00,$FF,$F0,$0F
5619  FEB5  C0 30 0C 03                .BYTE   $C0,$30,$0C,$03
5620                            ;
5621                            ; MASKTB: BIT MASK.    (ALSO PART OF DMASKTB! DO NOT SEPARATE)
5622                            ;
5623  FEB9  80 40 20 10        MASKTB: .BYTE   $80,$40,$20,$10,$08,$04,$02,$01
5624  FEBD  08 04 02 01
5625                            ;
5626                            ;
5627                            ;
5628                            ;
5629  FEC1  28 CA 94 46        COLRTB: .BYTE   $28,$CA,$94,$46,$00
5630  FEC5  00
5631                            ;
5632                            ;
5633                            ;
5634                            ;
5635                            ; CNTRLS: CONTROL CODES AND THEIR DISPLACEMENTS INTO THE
5636                            ; CONTROL CHARACTER PROCESSORS
5637                            ;
5638  FEC6  1B                  CNTRLS: .BYTE   $1B
5639  FEC7  79 F7                       .WORD   ESCAPE
```

```
5640  FEC9  1C          .BYTE   $1C
5641  FECA  7F F7       .WORD   CRSRUP
5642  FECC  1D          .BYTE   $1D
5643  FECD  8C F7       .WORD   CRSRDN
5644  FECF  1E          .BYTE   $1E
5645  FED0  99 F7       .WORD   CRSRLF
5646  FED2  1F          .BYTE   $1F
5647  FED3  AA F7       .WORD   CRSRRT
5648  FED5  7D          .BYTE   $7D
5649  FED6  B9 F7       .WORD   CLRSCR
5650  FED8  7E          .BYTE   $7E
5651  FED9  E6 F7       .WORD   BS
5652  FEDB  7F          .BYTE   $7F
5653  FEDC  10 F8       .WORD   TAB
5654  FEDE  9B          .BYTE   $9B
5655  FEDF  30 FA       .WORD   DOCRWS
5656  FEE1  9C          .BYTE   $9C
5657  FEE2  D4 F8       .WORD   DELLIN
5658  FEE4  9D          .BYTE   $9D
5659  FEE5  A4 F8       .WORD   INSLIN
5660  FEE7  9E          .BYTE   $9E
5661  FEE8  32 F8       .WORD   CLRTAB
5662  FEEA  9F          .BYTE   $9F
5663  FEEB  2D F8       .WORD   SETTAB
5664  FEED  FD          .BYTE   $FD
5665  FEEE  0A F9       .WORD   BELL
5666  FEF0  FE          .BYTE   $FE
5667  FEF1  6D F8       .WORD   DELCHR
5668  FEF3  FF          .BYTE   $FF
5669  FEF4  37 F8       .WORD   INSCHR
5670                    ;
5671                    ;
5672                    ;
5673                    ;
5674                    ;
5675                    ; ATAINT: ATASCI TO INTERNAL TABLE
5676                    ;
5677  FEF6  40 00 20 60 ATAINT: .BYTE   $40,$00,$20,$60
5678                    ;
5679                    ;
5680                    ; INTATA: INTERNAL TO ATASCI TABLE
5681                    ;
5682  FEFA  20 40 00 60 INTATA: .BYTE   $20,$40,$00,$60
5683                    ;
5684                    ;
5685                    ; ATASCI: ATASCII CONVERSION TABLE
5686                    ;
5687  FEFE  6C 6A 3B 80 ATASCI: .BYTE   $6C,$6A,$3B,$80,$80,$6B,$2B,$2A ;LOWER CASE
5688  FF02  80 6B 2B 2A
5689  FF06  6F 80 70 75         .BYTE   $6F,$80,$70,$75,$9B,$69,$2D,$3D
5690  FF0A  9B 69 2D 3D
5691  FF0E
5692  FF0E  76 80 63 80         .BYTE   $76,$80,$63,$80,$80,$62,$78,$7A
5693  FF12  80 62 78 7A
```

```
5694  FF16  34 80 33 36          .BYTE   $34,$80,$33,$36,$1B,$35,$32,$31
5695  FF1A  1B 35 32 31
5696  FF1E
5697  FF1E  2C 20 2E 6E          .BYTE   $2C,$20,$2E,$6E,$80,$6D,$2F,$81
5698  FF22  80 6D 2F 81
5699  FF26  72 80 65 79          .BYTE   $72,$80,$65,$79,$7F,$74,$77,$71
5700  FF2A  7F 74 77 71
5701  FF2E
5702  FF2E  39 80 30 37          .BYTE   $39,$80,$30,$37,$7E,$38,$3C,$3E
5703  FF32  7E 38 3C 3E
5704  FF36  66 68 64 80          .BYTE   $66,$68,$64,$80,$82,$67,$73,$61
5705  FF3A  82 67 73 61
5706  FF3E
5707  FF3E
5708  FF3E  4C 4A 3A 80          .BYTE   $4C,$4A,$3A,$80,$80,$4B,$5C,$5E ;UPPER CASE
5709  FF42  80 4B 5C 5E
5710  FF46  4F 80 50 55          .BYTE   $4F,$80,$50,$55,$9B,$49,$5F,$7C
5711  FF4A  9B 49 5F 7C
5712  FF4E
5713  FF4E  56 80 43 80          .BYTE   $56,$80,$43,$80,$80,$42,$58,$5A
5714  FF52  80 42 58 5A
5715  FF56  24 80 23 26          .BYTE   $24,$80,$23,$26,$1B,$25,$22,$21
5716  FF5A  1B 25 22 21
5717  FF5E
5718  FF5E  5B 20 5D 4E          .BYTE   $5B,$20,$5D,$4E,$80,$4D,$3F,$81
5719  FF62  80 4D 3F 81
5720  FF66  52 80 45 59          .BYTE   $52,$80,$45,$59,$9F,$54,$57,$51
5721  FF6A  9F 54 57 51
5722  FF6E
5723  FF6E  28 80 29 27          .BYTE   $28,$80,$29,$27,$9C,$40,$7D,$9D
5724  FF72  9C 40 7D 9D
5725  FF76  46 48 44 80           BYTE   $46,$48,$44,$80,$83,$47,$53,$41
5726  FF7A  83 47 53 41
5727  FF7E
5728  FF7E
5729  FF7E  0C 0A 7B 80          .BYTE   $0C,$0A,$7B,$80,$80,$0B,$1E,$1F ;CONTROL
5730  FF82  80 0B 1E 1F
5731  FF86  0F 80 10 15          .BYTE   $0F,$80,$10,$15,$9B,$09,$1C,$1D
5732  FF8A  9B 09 1C 1D
5733  FF8E
5734  FF8E  16 80 03 80          .BYTE   $16,$80,$03,$80,$80,$02,$1B,$1A
5735  FF92  80 02 1B 1A
5736  FF96  80 80 85 80          .BYTE   $80,$80,$85,$80,$1B,$80,$FD,$80
5737  FF9A  1B 80 FD 80
5738  FF9E
5739  FF9E  00 20 60 0E          .BYTE   $00,$20,$60,$0E,$80,$0D,$80,$81
5740  FFA2  80 0D 80 81
5741  FFA6  12 80 05 19          .BYTE   $12,$80,$05,$19,$9E,$14,$17,$11
5742  FFAA  9E 14 17 11
5743  FFAE
5744  FFAE  80 80 80 80          .BYTE   $80,$80,$80,$80,$FE,$80,$7D,$FF
5745  FFB2  FE 80 7D FF
5746  FFB6  06 08 04 80          .BYTE   $06,$08,$04,$80,$84,$07,$13,$01
5747  FFBA  84 07 13 01
```

```
5748                                ;
5749                                ;
5750                                ;
5751                                ;
5752                                ;
5753  FFBE  AD 09 D2        PIRQQ:  LDA     KBCODE
5754  FFC1  CD F2 02                CMP     CH1             ;TEST AGAINST LAST KEY PRESSED
5755  FFC4  D0 05                   BNE     PIRQ3           ;IF NOT, GO PROCESS KEY
5756  FFC6  AD F1 02                LDA     KEYDEL          ;IF KEY DELAY BYTE > 0
5757  FFC9  D0 20                   BNE     PIRQ4           ;IGNORE KEY AS BOUNCE
5758  FFCB  AD 09 D2        PIRQ3:  LDA     KBCODE          ;RESTORE AC
5759  FFCE  C9 9F                   CMP     #CNTL1          ;TEST CONTROL 1 (SSFLAG)
5760  FFD0  D0 0A                   BNE     PIRQ1
5761  FFD2  AD FF 02                LDA     SSFLAG
5762  FFD5  49 FF                   EOR     #$FF
5763  FFD7  8D FF 02                STA     SSFLAG
5764  FFDA  B0 0F                   BCS     PIRQ4           ;(UNCONDITIONAL) MAKE ^1 INVISIBLE
5765  FFDC  8D FC 02        PIRQ1:  STA     CH
5766  FFDF  8D F2 02                STA     CH1
5767  FFE2  A9 03                   LDA     #3
5768  FFE4  8D F1 02                STA     KEYDEL          ;INITIALIZE KEY DELAY FOR DEBOUNCE
5769  FFE7  A9 00                   LDA     #0              ;CLEAR COLOR SHIFT BYTE
5770  FFE9  85 4D                   STA     ATRACT
5771  FFEB  A9 30           PIRQ4:  LDA     #$30
5772  FFED  8D 28 02                STA     SRTIMR
5773  FFF0  68              PIRQ2:  PLA
5774  FFF1  40                      RTI
5775                                ;
5776                                ;
5777  FFF2  FF FF FF FF             .BYTE   $FF,$FF,$FF,$FF,$FF,$FF
5778  FFF6  FF FF
5779                                ;
5780  FFF8                  CRNTP8  =*
5781                                *=$FFF8
5782  FFF8  DD 57           CHKSUN. .DBYTE  $DD57
5783                                *=$14
5784  0014  00              KBDSPR  .BYTE   $FFF8-CRNTP8    ;^GDISPLC IS TOO LONG
5785  0015                          .END
```

ASSEMBLY ERRORS = 0

CROSS REFERENCE

LABEL	VALUE	REFERENCE
ACK	0041	-1658 1921
ACKREC	E9C6	1804 -1813
ADDCOR	030E	-501 2529 2533 2537
ADJ1	ED10	2547 -2551
ADJUST	ED08	2491 2494 -2546
ADRESS	0064	-270 4098 4123 4149 4151 4160 4161 4162
		4164 4230 4232 4236 4238 4241 4243 4274
		4333 4335 4404 4421 4423 4548 4551 4552
		4631 4633 4659 4664 4726 4733 4736 4738
		4740 4744 4812 4813 4816 4817 5070 5072
		5079 5082 5084 5127 5139 5141 5143 5146
		5149 5153 5154 5156 5157 5265 5356 5358
AFP	D800	-579
ALLPOT	D208	-673
ALLSEC	F30E	3864 -3869
ALOCAT	FE45	4124 -5540
ANCONV	FE69	4117 -5571
ANTIC	D400	-749 750 751 752 753 754 755 756
		757 758 759 760 761 762 763 764
APPEND	0001	-109
APPMHI	000E	-180 4739 4743
ASCC01	F705	4456 -4458
ASCZER	0030	-778 1209
ATACHR	02FB	-472 4286 4289 4294 4311 4350 4372 4395
		4399 4406 4459 4493 4504 4506 4584 5060
		5278 5505 5516 5520
ATAINT	FEF6	4321 -5677
ATAN	BE43	-613
ATASCI	FEFE	4458 -5687
ATEOF	F00B	3241 -3247
ATRACT	004D	-249 1359 1406 1412 1414 5770
AUDC1	D201	-681 2364 2374 2399
AUDC2	D203	-683 2375
AUDC3	D205	-685 2370
AUDC4	D207	-687
AUDCTL	D208	-688 2357
AUDF1	D200	-680 2318
AUDF2	D202	-682 2316
AUDF3	D204	-684 1768 2224 2616 3197
AUDF4	D206	-686 1770 2226 2618 3199
B192HI	0000	-1667 1769
B192LO	0028	-1666 1767
B600HI	0005	-1669 2225
B600LO	00CC	-1668 2223
BAD	EA63	1938 -1944
BADCOM	E9BF	1801 -1807 1828
BADDSK	F306	-3865 3901 3907
BADIOC	0086	-143 840
BADMOD	0091	-155 4119

BADST	EE9E	2919	2922	2968	-2970				
BEEP	F058	3167	3194	-3305					
BEEP1	F05A	-3306	3330						
BEGIN	ED14	2276	-2567	2579					
BELL	F90A	4380	-4704	5665					
BELL1	F90C	-4705	4707						
BFENHI	0035	-225	1793	1905	2002	2165	2205		
BFENLO	0034	-224	1790	1902	2005	2168	2200		
BITCLR	FB12	4602	5016	-5024					
BITCON	FAEB	-4997	5018	5024	5034				
BITQE1	FB31	5038	-5040						
BITQET	FB25	4596	4868	-5034					
BITMSK	006E	-277	5001	5020	5025	5037			
BITPUT	FB04	4689	-5016	5107	5116				
BITROL	FAFA	-5010	5137						
BITSET	FB06	4600	-5018						
BLACKB	F22A	-3717	3718						
BLFILL	EEC1	2997	-3000	3020					
BLIM	028A	-403	3180	3226	3245				
BLKB2	F230	3717	-3719						
BLKBDV	E471	-73	3460	3578	3580				
BLOAD	F36C	3906	-3911						
BOOT	F2CF	3687	-3840						
BOOTAD	0242	-365	3874	3876	3912	3915			
BOOT?	0009	-177	3842	3909	3969	3985			
BOTSCR	02BF	-432	4171	4179	4520	4526	4642	4931	5313
BPTR	003D	-235	3179	3216	3225	3229	3237	3254	3256
		3265	3284						
BRKABT	0080	-137	2646	3185	4444	4961			
BRKKEY	0011	-185	1356	1976	2084	2567	2589	2652	3186
		3211	3813	4445	4487	4962	4963		
BROKE	EDA4	1978	2086	2569	2590	-2641			
BS	F7E6	-4569	5651						
BS1	F80D	4571	-4586						
BS2	F805	4579	4581	-4583					
BS3	F7F5	4574	-4576						
BSA	F7EC	-4572							
BUFADR	0015	-188	2835	2844	2848	2869	2871		
BUFCNT	006B	-275	4364	4388	4390	5197	5207	5210	5218
		5227							
BUFFH	0004	-3398	3399	3861					
BUFFL	0000	-3399	3859						
BUFFUL	EECB	2994	-3005						
BUFRFL	0038	-228	2074	2131	2181				
BUFRHI	0033	-223	1792	1904	1999	2001	2162	2164	2203
		2633	2635						
BUFRLO	0032	-222	1788	1900	1970	1997	2004	2029	2153
		2160	2167	2198	2624	2626	2630	2632	
BUFSTR	006C	-276	4367	4369	4384	4386	4562	4566	4905
		5192	5194	5241	5243				
CAINI	F239	3646	-3727						
CART	BFFC	-3413	3635	3643	3758	3760	3761	3771	
CARTAD	BFFE	-3415	3727	3728	3766				

CARTCS	BFFA	-3412	3697	3703					
CARTFG	BFFD	-3414	3679	3682	3694	3700	3763		
CAS31	EC62	2373	-2378						
CASBUF	03FD	-548	3107	3108	3228	3239	3244	3255	3286
		3292	3370	3372	3373	3398	3399	3870	3878
		3880	3883						
CASENT	EB84	1753	-2218						
CASET	0060	-1637	1751	2310	2372				
CASETV	E440	-55	3122	3490	3827				
CASFLG	030F	-502	1758	2070	2255				
CASINI	0002	-170	3987	3989	3992				
CASORG	EF41	-26	3102	3140					
CASRED	EBB7	2219	-2254						
CASSBT	004B	-246	3866	3900	3903	3953	3979	3983	
CASSET	0043	-123							
CASSPR	0014	-3379							
CAUX1	023C	-352	1782						
CAUX2	023D	-353	1784						
CBAUDH	02EF	-460	2611	2617	3148				
CBAUDL	02EE	-459	2610	2615	3146				
CBINI	F23C	3638	-3728						
CBUFH	0003	-3107	3108	3347					
CBUFHI	0002	-1687	1688	1791					
CBUFL	00FD	-3108	3349						
CBUFLO	003A	-1688	1787						
CCOMND	023B	-351	1779						
CDEVIC	023A	-350	1687	1688	1776				
CDTMA1	0226	-331	1540	2662	2664				
CDTMA2	0228	-332	1541						
CDTMF3	022A	-333	1467	3174	3176	3210	3213		
CDTMF4	022C	-335							
CDTMF5	022E	-337							
CDTMV1	0218	-324	1463	1464	1547	1549	1551	1552	1554
		1574	1576						
CDTMV2	021A	-325							
CDTMV3	021C	-326							
CDTMV4	021E	-327							
CDTMV5	0220	-328							
CDUBL	EF26	3077	-3080						
CH	02FC	-473	1490	4065	4440	4447	4452	5765	
CH1	02F2	-464	5754	5766					
CHACT	02F3	-466	1454	4088					
CHACTL	D401	-751	1455						
CHAR	02FA	-471	4280	4322	4324	5047	5057		
CHBAS	02F4	-467	1452	4086					
CHBASE	D409	-757	1453						
CHKDON	EABC	2015	-2022	2037					
CHKERR	008F	-152	2138						
CHKSNT	003B	-231	1966	2008	2014	2046			
CHKSUM	0031	-221	1965	1974	2011	2033	2035	2073	2135
		2156	2158	2628					
CHKSUN	FFF8	-5782							
CHKTIM	EAFB	-2084	2091						

```
CHRORG    E000          -18
CICLO2    E53F          915  -917
CICLOS    E533          867  -912
CIERR1    E4D1          835  -840
CIERR2    E6B0         1189 -1199
CIERR3    E50F         -883
CIERR4    E511          856  -884   888   894   934
CIJUMP    E693         1168 -1174
CINI      F3E1         3972 -3992
CIO       E4C4          785  -829
CIOCHR    002F         -218   829   965   970   980   999  1003  1033
                       1040  1051  1099
CIOI1     E4A8         -801   812
CIOINT    E4A6          789  -800  1228
CIOINV    E46E          -72   788  3828
CIOORG    E4A6          -21   797
CIOPEN    E509          865  -878
CIOSPR    0014        -1231
CIOV      E456          -64   784  3660  3945
CIREAD    E569          871  -948
CIRT3     E62B        -1090  1095
CIRTN1    E61B          841   884   954  1023 -1079
CIRTN2    E61D          907   923   944   966  1013  1074 -1084
CIST1     E559          930  -937
CISTSP    E54E          869  -928
CIWRIT    E5C9          872 -1017
CIX       00F2         -629
CKEY      004A         -245  3834  3975  3984
CKSTC     EE11         2814 -2816
CLICK     FCD8         4453  4705 -5333
CLICK1    FCDA        -5334  5337
CLOSE     000C           -89
CLOSEC    F02B         3123 -3277
CLRCHP    F284        -3802  3807
CLRCOD    007D        -4008  4290
CLRLI1    FBA6        -5127  5129
CLRLIN    FB9B         4675 -5122
CLRRAM    F140        -3568  3571  3575
CLRSC2    F7BF        -4548  4550  4554
CLRSC3    F7CE        -4556  4559
CLRSCR    F7B9         4258  4292 -4545  5649
CLRTAB    F832        -4601  5661
CLRTBS    F430        -4103  4105
CLS       007D        -3393  3504
CLWRT     F038         3278 -3284
CMODE     EF1E        -3076  3092
CNTL1     009F        -4009  5759
CNTRLS    FEC6         4420  4422  5277 -5638
COLAC     0072         -280  5418  5473  5475  5476  5478  5482
COLBK     D01A         -722
COLCR     FCE4         4560  4879 -5343
COLCR1    FCEE         5345 -5348
COLCR2    FCF0         5347 -5349
```

```
COLCRS    0055        -260  4368  4387  4530  4531  4536  4538  4539
                      4565  4572  4577  4588  4657  4697  4756  4758
                      4789  4794  4823  4825  4838  4840  4841  4853
                      4881  4948  4953  4956  5123  5180  5195  5204
                      5223  5349  5378  5380  5487  5488  5491  5493
                      5495  5497
COLDST    0244        -366  3542  3583  3691
COLDSV    E477         -75  3468
COLINC    007A        -285  5384  5406  5485
COLON     003A        -779
COLOR0    02C4        -439  4108
COLOR1    02C5        -440
COLOR2    02C6        -441
COLOR3    02C7        -442
COLOR4    02C8        -443  4147
COLPF0    D016        -718
COLPF1    D017        -719
COLPF2    D018        -720
COLPF3    D019        -721
COLPM0    D012        -714  1449
COLPM1    D013        -715
COLPM2    D014        -716
COLPM3    D015        -717
COLRSH    004F        -251  1418  1447
COLRTB    FEC1        4107 -5629
COLUMN    FE8D        4843  4955 -5594
COM1      E647        1111 -1118
COM2      E662        1115 -1132
COMENT    E63D         893   902   914   937   957  1026 -1109
COMFRM    E978       -1767  1808
COMMND    E974       -1762  1875
COMPLT    0043       -1660  1923
COMPUT    ECA7       -2489  2609
COMRE1    F7A7        4534 -4537  4541  4542
COMRET    F789        4519 -4523  4527
COMTAB    E6C9         864  1123 -1225
CONSOL    D01F        -727  1444  3311  3316  3831  5334
CONTIN    EC35        2328 -2352
CONVR1    F97E       -4783  4787
CONVR2    F988        4784 -4788
CONVR3    F98F       -4791  4797
CONVR4    F99C        4792 -4798
CONVR5    F9A6        4802 -4804
CONVR6    F9A7       -4805  4808
CONVRT    F947        4273  4323  4630  4658  4698 -4754  5124  5260
COS       BD73        -612
COUNT     ED3F       -2589  2600  2603  2613
COUNTR    007E        -289  5431  5441  5449  5450  5523  5525  5526
                      5528
CR        009B        -129  2996  3504  3516  3528  4295  4373  4398
                      4488
CRETRI    000D       -1690  1762  1821
CRETRN    EBE3        2248 -2280
```

```
CRETRY    0036          -226  1763  1807  1822
CRITIC    0042          -240  1423  1748  1882
CRLOOP    F5E7         -4307  4310
CRNTP1    E6D5         -1229  1231
CRNTP2    E944         -1608  1610
CRNTP3    EDEA         -2733  2735
CRNTP4    EE78         -2878  2881
CRNTP5    EF41         -3098  3102
CRNTP6    F0E3         -3377  3379
CRNTP7    F3E4         -3998  4001
CRNTP8    FFF8         -5780  5784
CRSINH    02F0          -462  1358  4100  4343
CRSRDN    F78C         -4524  5643
CRSRL1    F7A3          4532 -4535
CRSRLF    F799         -4530  4576  5215  5222  5645
CRSROR    008D          -150  4967
CRSRRT    F7AA         -4538  4587  5647
CRSRUP    F77F         -4518  4582  5641
CSBOOT    F3B2          3671 -3967
CSBOT2    F3C0          3968 -3975
CSIDE     EF2E          3081 -3084
CSIO      F0AC         -3351
CSOPIV    E47D           -77  3136  3980
CSTAT     0288          -401
CTIA      D000          -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   734
                         735   736   737   738   739   740   741   742
                         743   744   745   746   747
CTIMHI    0000         -1693  2467
CTIMLO    0002         -1692  2466
CTRLC     0092         -3394
D         0044         -2906  3080
DAUX1     030A          -497  1781  2965  3089  3858  3896
DAUX2     030B          -498  1783  2231  2258  2280  3364  3856
DBDDEC    F913          4166 -4718
DBDEC     F91F          4156 -4729
DBSECT    0241          -364  3894
DBSUB     F921          4126  4143  4719 -4730
DBSUB1    F934          4737 -4739
DBUFHI    0305          -492  2202  2821  2870  3040  3348  3862
DBUFLO    0304          -491  2197  2819  2868  3039  3350  3860
DBUFSZ    0014         -2900  3082
DBYTHI    0309          -496  2204  2826  2856  3054  3344
DBYTLO    0308          -495  2199  2824  2854  3052  3346
DCB       0300          -486
DCOMND    0302          -489  1778  2805  2812  2830  2837  2964  3046
                        3088  3342  3357  3850  3957
DCTIM1    E902          1548 -1552
DCTIMR    E8F5          1420  1457  1466 -1547
DCTXF     E90F          1550  1553  1555 -1558
```

```
DDEVIC    0300          -487  1750  1773  2309  2371  2803  3042  3201
                        3352
DECBF1    E66D          1140 -1142
DECBFL    E663           983   987  1054  1058 -1137
DEGFLG    00FB          -634
DEGON     0006          -637
DELAY0    EC90         -2458  2462
DELAY1    EC92         -2459  2460
DELCH1    F870         -4630  4647
DELCH2    F896          4640  4643 -4648
DELCHR    F86D         -4629  5667
DELETE    0021           -98
DELLI1    F8DD         -4681  4692
DELLI2    F8F8         -4696
DELLIA    F8D7         -4678
DELLIB    F8DB         -4680  4702  5269
DELLIN    F8D4         -4677  5657
DELTAC    0077          -283  5400  5403  5407  5409  5410  5412  5413
                        5415  5425  5430  5434  5474  5477
DELTAR    0076          -282  5388  5392  5396  5435  5437  5455
DELTI1    FC82         -5265  5268
DELTI2    FC89         -5269
DELTI3    FC8C          5259  5266 -5270
DELTIA    FC68          4651 -5252
DELTIB    FC70          5254 -5256
DELTIM    FC73          4575 -5257
DERR      E9F6          1840 -1861
DERR1     EA06          1810  1849 -1872
DERR5     F10D         -3516  3520  3521
DERRH     00F1         -3520  3521  3927
DERRL     000D         -3521  3926
DERROR    0090          -154  1929
DEVS1     E6A6         -1191  1196
DEVS2     E6B5          1192 -1204
DEVS3     E6C5          1211 -1213
DEVS4     E6C8          1201 -1217
DEVSRC    E69E           887   933 -1187
DFLAGS    0240          -363  3871
DHLINE    FE81          4782 -5587
DIGRT     00F1          -628
DINDEX    0057          -261  4084  4116  4172  4203  4341  4781  4842
                        4855  4883  4934  4941  5048  5346
DINI      F37E          3845  3908 -3919
DINIT     EDEA          2765 -2794
DIRECT    0002          -110
DISK      0044          -125
DISKID    0031         -2752  2802
DISKIV    E450           -62
DISKM     F3A4          3954 -3956
DISPLA    E410         -4034
DISPLY    0053          -121
DIV2TB    FEA5          4788 -5611
DLISTH    D403          -753  1436
```

```
DLISTL    D402          -752  1438
DMACTL    D400          -750  1440
DMASK     02A0          -419  4275  4329  4331  4820
DMASKT    FEB1          4819 -5618
DNACK     008B          -148  1933
DOB1      FC1A          5200 -5202
DOBOOT    F2ED          3852 -3855  3867
DOBU1A    FC29          5203  5206 -5209
DOBUF1    FC12         -5198  5213
DOBUF2    FC39          5208 -5216  5228
DOBUF3    FC51          5225 -5227
DOBUF4    FC55          5217  5221 -5229
DOBUFC    FC00          4383 -5189
DOCR      FA34          4591  4856  4861  4864 -4879
DOCR1     FA00          4852 -4855
DOCR1A    FA29          4869  4871 -4874
DOCR1B    FA14          4859 -4864
DOCR2     FA3D         -4883
DOCR2A    FA4A          4886 -4890
DOCR2B    FA4D          4889 -4891
DOCR4B    FA61          4901 -4903  4909
DOCRWS    FA30          4297  4397  4862 -4877  5655
DOINTP    ECE9         -2523  2526
DOLC01    FBE5         -5169  5177
DOLC02    FBF8          5171 -5178
DOLCOL    FBDD          4537  4586  4598  4626  4653  4676  4677  4703
                        4874  4912  5160 -5165  5239  5256
DOPEN     F3F6          4034 -4077
DOPEN1    F460         -4125  4128
DOPEN2    F4D5         -4181  4184
DOPEN3    F4FA         -4199  4202
DOPEN4    F51C         -4215  4218
DOPEN5    F524          4205 -4219
DOPEN7    F588          4257 -4262
DOPEN8    F438         -4107  4110
DOPEN9    F577          4249 -4255
DOPENA    F457          4118 -4121
DOSINI    000C          -179  3879  3881  3919  3986  3988
DOSS      F6AD          4375 -4409
DOSVEC    000A          -178  3579  3581  3707
DOUBLE    0044         -1652
DRAW      FCFC          4039 -5366
DRAW1     FD37          5389 -5397
DRAW10    FE42          5452  5529 -5531
DRAW11    FD99          5451 -5453
DRAW2     FD5C          5404  5414 -5416
DRAW3     FD83          5433  5436 -5440
DRAW3A    FD62         -5419  5424
DRAW4A    FD90         -5449  5530
DRAW5     FDA4          5457 -5459
DRAW5A    FDB2          5462 -5466
DRAW6     FDBE          5461  5465 -5472
DRAW6A    FDD7          5481 -5485
```

```
DRAW6B    FDEB          5486 -5495
DRAW7     FDF1          5490  5492  5494  5496 -5498
DRAW8     FDF6          5480  5484 -5500
DRAW8A    FE0A         -5507  5518
DRAW8B    FE27          5514 -5519
DRAW8C    FE13         -5512
DRAW9     FE30          5503 -5522
DRAWA     FD0C          5369 -5375
DRAWB     FD0B          5371 -5374
DRAWLN    0011           -95
DRETRI    0001         -1691  1760
DRETRY    0037          -227  1761  1872
DRKMSK    004E          -250  1417  1448
DSKFMS    0018          -192
DSKIF     EDF0          2766 -2802
DSKINV    E453           -63  3851  3960 (2766)
DSKORG    EDEA           -24  2735  2775
DSKRDE    F381          3865  3899 -3926
DSKSPR    0014         -2881
DSKTIM    0246          -370  2795  2804  2836
DSKUTL    001A          -193
DSPFLG    02FE          -475  4415
DSTAT     004C          -247  4091  4120  4248  4347  4349  4371  4392
                        4401  4486  4723  4731  4747  4960  4968
DSTATS    0303          -490  1813  1845  1884  2218  2823  2857  3050
                        3362
DT1       00FA         -3114  3242  3287
DTA       00FC         -3113  3262
DTIMLO    0306          -493  2417  2809  3056  3356
DUNIT     0301          -488  1774  3044  3354  3848  3959
DUNUSE    0307          -494
DVSTAT    02EA          -458  2744  2745  2944  3064
EDITOR    E400         -4021
EDITRV    E400           -51  3492  3719  3721  3823  4017
EEXP      00ED          -624
EGETC1    F650         -4370  4381
EGETC2    F66E          4374 -4382
EGETC3    F67C          4365 -4388
EGETC5    F691          4389  4391 -4397
EGETC6    F66B          4379 -4381
EGETC7    F680         -4390  4393
EGETCH    F63E          4023 -4362
ENBOOT    F35A          3895 -3903
ENDACK    F1BD          3642  3644 -3651
ENDBCK    F1AC          3634  3636 -3640
ENDDIF    EE69          2839 -2857
ENDPT     0074          -281  4919  4922  5432  5442  5444  5460  5464
                        5479  5483
ENDRAM    F280          3784 -3792
ENINTP    ECEF         -2528
ENSPEC    F256          3759  3762  3765 -3771
ENTVEC    002C           -824
EOF       0088         -3395
```

```
EOFERR    0088          -145  3248  4485
EOL       009B          -780   981  1004  1052  1069
EOPEN     F3FC          4021 -4080  4251  4936
EOT       00FE         -3115  3240  3295
EOUTC5    F6BE          4411 -4415
EOUTC6    F6B5         -4412  4417
EOUTCH    F6A4          4024 -4406
ERANGE    FA88          4363  4408 -4931
ERETN     F6BB         -4414
ERRFLG    023F          -361  1800  1837  1848  1861  1896  1941
ERROR     0045         -1661  1926
ERRTN     E4C0          -816   817   818
ERRTNH    00E4           805  -817   818   919
ERRTNL    00C0           803  -818   921
ESCAPE    F779         -4515  5639
ESCFLG    02A2          -421  4412  4416  4418  4516
ESIGN     00EF          -626
ESTSCM    F173          3584 -3599
EXP       DDC0          -604
EXP10     DDCC          -605
EXTEN1    FB7E         -5100  5111
EXTEN3    FB8B         -5108
EXTEN4    FB92          5109 -5112
EXTEND    FB7B          4655 -5098
FADD      DA66          -590
FASC      D8E6          -583
FCAX      F032         -3281  3289  3297
FCHRFL    00F0          -627
FDIV      DB28          -592
FEOF      003F          -237  3165  3223  3247
FILDAT    02FD          -474  5515
FILFLG    02B7          -425  5375  5502
FILLBF    EEC3         -3001  3004
FILLIN    0012           -96
FINDX     ECDA         -2511
FLDOP     DD8D          -594
FLDOR     DD89          -593
FLD1P     DD9C          -596
FLD1R     DD98          -595
FLOPPY    0030         -1634
FLPTR     00FC          -638
FMOVE     DDB6          -599
FMSZPG    0043          -242
FMTD      EE4A         -2840
FMUL      DADB          -591
FNCNOT    0092          -156  1167
FOMAT     0021         -2755  2806  2838
FOOEY     EAE0          2047 -2056
FORMAT    0022           -99
FPI       D9D2          -588
FPREC     0006          -575   648   649
FPSCR     05E6          -648   649   650
FPSCR1    05EC          -649   651
```

```
FPTR2     00FE          -639
FR0       00D4          -619
FR1       00E0          -621
FR2       00E6          -622
FRE       00DA          -620
FREQ      0040          -238  3305  3322
FRMADR    0068         -4011  4660  4665  5076  5083  5085
FRMERR    008C          -149  2122
FRX       00EC          -623
FSCR      05E6          -650
FSCR1     05EC          -651
FSTOP     DDAB          -598
FSTOR     DDA7          -597
FSUB      DA60          -589
FTYPE     003E          -236  3155  3363  3978
GBX       EFE8         -3231  3235
GBYTE     EFD6          3123 -3223  3246
GETCAR    0007         -3388
GETCH     F593          4036 -4268  4394
GETCHR    0007           -86
GETDAT    0040         -2757  2810
GETOUT    F749          4443 -4488
GETPLT    F5A2          4269 -4273  4339  4604  4617  4644  5216  5513
GETREC    0005           -85
GETSEC    F39D          3863  3897  3905 -3953
GLBABS    02E0          -450
GOBACK    E80B         -2092
GOERR     EF3D          3085 -3091
GOHAND    E689           897   916   938   964   969   998  1041  1070
                       -1167
GOOD      EA65          1922  1924  1942 -1946
GOODST    EE32          2828 -2830
GOON      EB68          2172 -2180
GOREAD    ED73          2606 -2615
GPRIOR    026F          -374  1441  4129  4148
GRACTL    D01D          -725
GRAFM     D011          -713
GRAFP0    D00D          -709
GRAFP1    D00E          -710
GRAFP2    D00F          -711
GRAFP3    D010          -712
HARDI     F281          3560 -3800
HATABS    031A          -515   516  1118  1120  1191  3619
HDR       00FB         -3116
HITCLR    D01E          -726
HITIMR    ECD0         -2505  2508
HITONE    0005         -1670  2317
HOLD1     0051          -256  4121  4199  4215  4223  4678  4768  4776
                        5168  5169  5176  5240
HOLD2     029F          -418  4770  4779
HOLD3     029D          -416  4893  4906  4910
HOLD4     02BC          -429  5506  5519
HOLD5     02BD          -430
```

```
HOLDCH    007C          -287  4450  4490  4500
HOME      F7D6         -4560  4966
HOWMCH    F263         -3777  3791
HPOSM0    D004          -700
HPOSM1    D005          -701
HPOSM2    D006          -702
HPOSM3    D007          -703
HPOSP0    D000          -696
HPOSP1    D001          -697
HPOSP2    D002          -698
HPOSP3    D003          -699
HSCROL    D404          -754
ICAX1     034A          -531  3659
ICAX1Z    002A          -214   949  1018  3156  4080  4082  4175  4255
                        4441
ICAX2     034B          -532
ICAX2Z    002B          -215  3075  3154  4077
ICBAH     0345          -526  1087  3657  3940
ICBAHZ    0025          -209  1088
ICBAL     0344          -525  1085  3655  3938
ICBALZ    0024          -208   973  1039  1086  1147  1149  1188  1208
ICBLH     0349          -530  1159
ICBLHZ    0029          -213  1161
ICBLL     0348          -529  1156  3944
ICBLLZ    0028          -212   961   962  1030  1031  1034  1137  1138
                        1141  1142  1157  1158  1160
ICCOM     0342          -523  3653  3942
ICCOMT    0017          -190   863   901  1122
ICCOMZ    0022          -206   854   948   975   993  1017  1046  1064
                        5367
ICDNO     0341          -522
ICDNOZ    0021          -205  1213
ICHID     0340          -521   802   942
ICHIDZ    0020          -204   878   918   928   943  1109  1205
ICIDNO    002E          -217   830   941  1084  1100  1154  1180
ICPTH     0347          -528   806
ICPTHZ    0027          -211   906   920
ICPTL     0346          -527   804
ICPTLZ    0026          -210   904   922
ICSPR     034C          -533
ICSPRZ    002C          -216   217   218   824   903   905  1119  1121
                        1125  1128  1129  1130  1175  1177
ICSTA     0343          -524
ICSTAZ    0023          -207   913  1009  1079  1101  1169
IDENT     F0F2         -3504  3512  3513  3515
IDENTH    00F0         -3512  3513  3711
IDENTL    00F2         -3513  3710
IFP       D9AA          -584
IHINIT    E6D5          1242 -1281
INATA1    FB4A          5050 -5060
INATAC    FB32          4270 -5047
INBUFF    00F3          -630
INC2A     F9F8          4844  4849 -4851
```

```
INCBF1    E676          1148 -1150
INCBFP    E670           974  1043 -1147
INCRS1    FA77          4863  4892  4895  4897 -4912
INCRS2    F9E4          4839 -4841
INCRS3    F9F7          4846  4848 -4850  4854
INCRSA    F9DC          4611 -4837  5509
INCRSB    F9D4          4271  4637 -4833  5209
INCRSC    F9DA          4834 -4836
INCRSR    F9D8          4300 -4835
INIMLH    0007         -3391  3821
INIMLL    0000         -3390  3819
INIT      EF41          3126 -3145
INSCH1    F852         -4615
INSCH3    F85E          4614 -4621
INSCH4    F844         -4608  4620
INSCH5    F86A          4623 -4626
INSCH6    F861         -4622  4625
INSCHR    F837         -4603  5669
INSCLR    0020          -115
INSDAT    007D          -288  4605  4615  4618  4836  4860  4870  4878
                        4896
INSLI1    F8C6         -4671  4674
INSLI2    F8CE          4670 -4675
INSLIA    F8A5         -4655  4873
INSLIN    F8A4         -4654  5659
INTABS    0200          -311  1244  1713  3608  4059
INTATA    FEFA          5059 -5682
INTEMP    022D          -336  1569  1573
INTINV    E46B           -71  1241  3830
INTORG    E6D5           -22  1231  1280
INTSPR    0014         -1610
INTTBL    EC88         -2440
INTZBS    0010          -183  3593
INVFLG    02B6          -424  4464  4466  4505
IOC1      E4D6           837  -844
IOC1A     E4D8          -845   850
IOC2      E4F3           861  -863
IOC6      E514           880  -887
IOC7      E519         -893
IOCB      0340          -520   845  1091
IOCBAS    0020          -203   846  1090
IOCBSZ    0010          -201   202   809
IOCFRE    00FF          -104   801   879   917   929
IRQEN     D20E          -692  1299  1301  1307  1309  1314  1317  1319
                        1321  1325  1327  1332  1334  1339  1341  1343
                        1346  1348  1352  1354  2020  2054  2353  2395
                        4095
IRQST     D20E          -678  1296  1304
ISEOF     F00D          3224 -3248
ISRODN    EA90          1716 -1994  2441  2446  2447
ISRSIR    EB11          1715 -2111  2440  2444  2445
ISRTD     EAD1          1717 -2046  2442  2448  2449
JSRIND    F6A1         -4404  4424
```

```
JTADRH    00EB        -2293  2294  2663
JTADRL    00F0        -2294  2661
JTIMER    E8F0        -2292  2293  2294
JTIMR1    E8EF         1422 -1540
JTIMR2    E8F2         1459 -1541
K1        F729         4469 -4473
K2        F734         4474 -4478
K3        F73F         4479 -4483
K4        F776         4503 -4507
K5        F768         4492  4495  4497  4499 -4502
K6        F74D         4484 -4490
K7        F745         4446 -4486
K8        F773         4489 -4506
KBCODE    D209         -674  1489  5753  5758
KBD       004B         -120
KBDHND    E420        -4048
KBDORG    F3E4          -28  4001  4062
KBDSPR    0014        -5784
KEYBDV    E420          -53  3334  3336  3496  3825  4043
KEYDEL    02F1         -463  1475  1477  5756  5768
KGETC1    F71E         4463 -4468
KGETC2    F6DD        -4439  4461  4467  4472  4477  4482
KGETC3    F6FE        -4454  4501
KGETCH    F6E2         4050  4370 -4441  4449
LBFEND    05FF         -652
LBPR1     057E         -644
LBPR2     057F         -645
LBUFF     0580         -646   647
LDPNTR    EB6E         1824  1857 -2196  2244  2271
LEDGE     0002         -253  3599
LENGTH    022F        -1228
LFRTCM    F7A5        -4536  4544
LINBUF    0247         -372  5069  5071
LINZBS    0000         -167
LIRG      0000        -3396
LMARGN    0052         -257  3600  4260  4533  4543  4570  4573  4589
                       4656  4696  5122  5212  5220  5242  5253  5258
                       5263  5348
LO1GET    FB22        -5032  5101  5170
LO2GET    FB23         4683 -5033
LOCKFL    0023         -100
LOG       DECD         -606
LOG10     DED1         -607
LOGCOL    0063         -269  4377  4561  4569  4595  4599  4601  4609
                       4613  4635  4639  4837  4857  5166  5172  5175
                       5179  5181  5190  5211  5219  5230  5252  5257
LOGGET    FB20         4592  4700 -5031
LOGMAP    02B2         -423  4556  4693  4695  4907  5010  5011  5012
LOTONE    0007        -1671  2315
LPENH     0234         -344  1434
LPENV     0235         -345  1432
MOPF      D000         -728
MOPL      D008         -736
```

```
M1PF        D001          -729
M1PL        D009          -737
M2PF        D002          -730
M2PL        D00A          -738
M3PF        D003          -731
M3PL        D00B          -739
MASKTB      FEB9          5000 -5623
MAXDEV      0021          -516  1110  1190
MAXIOC      0080          -202   811   836
MEMLO       02E7          -456  3820  3822
MEMORY    M 0000             0
MEMTOP      02E5          -455  3816  3818  4066  4237  4239
MLTTMP      0066          -271   272  4012  4762  4764  4765  4767  4769
                          4772  4774  4777  4778  4780  4785  4786  4800
                          4801  4803  4810  4815
MODATA      E9F0          1846 -1857
MODEM       004D          -124
MONORG      F0E3           -27  3379  3480
MONSPR      0014         -4001
MOTRGO      0034         -1682  2238  2265  3169  3203
MOTRST      003C         -1683  1725  2283  2642  3281
MOVLI1      FB58         -5074  5081
MOVLI2      FB7A          5090 -5092
MOVLIN      FB4E          4671 -5069
MOVVEC      F17D         -3607  3610
MVBUFF      F32D         -3882  3898
MVNXB       F32F         -3883  3886
MXDMDE      FE5D          4194 -5550
MXDMOD      0010          -114
N           004E         -2905  3076  3091
NACK        004E         -1659
NARG        0000             0
NBUFSZ      0028         -2899  3078
NCOMHI      003C         -1681  1728  2082
NCOMLO      0034         -1680  1795
NEWCOL      0061          -268  5379  5381  5398  5401
NEWROW      0060          -267  5377  5386
NLR         F005          3243 -3245
NMIEN       D40E          -762  1282  1571  1578
NMIRES      D40F          -763  1395
NMIST       D40F          -764  1383  1387  1579
NOA1        F1F1          3678 -3680
NOA2        F212          3693  3696 -3698
NOB1        F1F8          3681 -3683
NOBOOT      F1FF          3684 -3690
NOCAR2      F220          3699 -3707
NOCART      F1FC          3676 -3687  3702
NOCKSM      003C          -232  1908  2171  2175
NOCLR       EAEB          2071 -2074
NOCSB2      F3BF          3971 -3973
NOCSBT      F3E0          3976 -3990
NODAT       0000         -2756
NOFUNC      F63D          4026  4051  4053 -4351
```

```
NOINIT    F2DC          3844 -3846
NOISE1    EC49          2362 -2364  2367
NOKEY     F2CE          3833 -3835
NOMOD     F4AB          4158 -4162
NONDEV    0082          -139  1199
NORMAL    004E         -1651
NOROWS    FE99          4890  4942 -5602
NOSCR1    FA32          4876 -4878
NOSCRL    FA2C         -4875
NOT8      F48B          4134  4145 -4148
NOTCAS    EC10          2311 -2320
NOTCST    E96B          1752 -1757
NOTDER    EA52          1927 -1933
NOTDON    EA81         -1976  1981
NOTE      0026          -103
NOTEND    EAC0          2003  2006 -2028
NOTERR    EA00          1862 -1868
NOTMXD    F4F5          4174  4177 -4197
NOTOPN    0085          -142   819  1114
NOTYET    EB3E          2132 -2151
NOWARM    F2DD          3841 -3847
NOWRP0    EA98          1998 -2001
NSIGN     00EE          -625
NTBRK0    EA88          1977 -1980
NTBRK1    EB02          2085 -2088
NTBRK2    ED1B          2568 -2571
NTFRAM    EB1F          2120 -2125
NTOVRN    EB27          2126 -2131
NTWRP1    EB52          2161 -2164
NUMDLE    FE51          4194  4198 -5547
NVALID    0084          -141   853  5372
NWOK      EA56          1915  1931 -1936
NXTENT    F18C         -3618  3621
ODNHI     00EA         -2446  2447
ODNLO     0090         -2447
OFFCRS    FAE4          4382  4409 -4980
OKTIM1    ED23          2574 -2577
OKTIMR    ED4C          2593 -2597
OLDADR    005E          -266  4814  4818  4982
OLDCHR    005D          -265  4340  4981
OLDCOL    005B          -264  5399  5402
OLDROW    005A          -263  4308  5387  5421
OPEN      0003           -84   855
OPENC     EF4C          3123 -3154
OPINP     EF5D          3137  3159 -3163
OPNCOM    F404          4079 -4084
OPNEDT    F118         -3528  3530  3531
OPNERR    F453         -4119
OPNH      00F1         -3530  3531  3656
OPNIN     0004          -111   113
OPNINO    000C          -113
OPNL      0018         -3531  3654
OPNOT     0008          -112   113
```

```
OPNOUT    0002          -2898
OPNRTN    EF8F           3168 -3187  3195
OPNTMP    0066           -272  4131  4140  4169  4195  4196  4197
OPOK      EFD3           3181 -3217
OPOUT     EF95           3161 -3191
OPSYS     F17B          -3606
OSRAM     F294           3611 -3813
OUTCH     F5B7           4037 -4286
OUTCH2    F5FF          -4322  4346  4608
OUTCHA    F5BD          -4289
OUTCHB    F5D7           4296 -4299
OUTCHE    F5CA           4291 -4294  4413
OUTPLT    F5E0           4299 -4304  4305  4585  5501  5517
OVRRUN    008E           -151  2128
POPF      D004           -732
POPL      D00C           -740
P1PF      D005           -733
P1PL      D00D           -741
P2PF      D006           -734
P2PL      D00E           -742
P3PF      D007           -735
P3PL      D00F           -743
PACTL     D302           -768  1284  1290  1362  1726  2239  2266  2284
                        2643  3170  3204  3282
PADDL0    0270           -376  1512
PADDL1    0271           -377
PADDL2    0272           -378
PADDL3    0273           -379
PADDL4    0274           -380  1514
PADDL5    0275           -381
PADDL6    0276           -382
PADDL7    0277           -383
PAGETB    FE75           4157 -5579
PBCTL     D303           -769  1285  1291  1366  1729  1796  2083  2644
PBPNT     001D           -196  2979  2989  2995  3006  3019
PBRK      EF8B          -3185  3212
PBUFSZ    001E           -197  2960  2993  3003  3051  3087
PBYTE     F010           3123 -3254
PCOLR0    02C0           -435  1446
PCOLR1    02C1           -436
PCOLR2    02C2           -437
PCOLR3    02C3           -438
PDEVN     0040          -2902  3041
PENH      D40C           -760  1433
PENV      D40D           -761  1431
PHACR1    FC9F          -5291  5294
PHACRS    FC9D           4603  4629  5189 -5290  5504
PHCHLO    EE7F          -2945  3007  3008
PHCLOS    EEDC           2918 -3018
PHINIT    EE78           2923 -2937
PHOPEN    EE9F           2917 -2977
PHPUT     EF14           2969 -3064
PHSTAT    EE81           2921 -2959  2977
```

PHSTLO	EE7D	-2944	2961	2962					
PHWRIT	EEA7	2920	-2987						
PIA	D300	-765	766	767	768	769			
PIRQ	E6F3	-1293	1603	1604					
PIRQ1	FFDC	5760	-5765						
PIRQ2	FFF0	-5773							
PIRQ3	FFCB	5755	-5758						
PIRQ4	FFEB	5757	5764	-5771					
PIRQH	00E6	1264	-1603	1604					
PIRQL	00F3	1266	-1604						
PIRQQ	FFBE	4060	-5753						
PLACR1	FCAA	-5301	5304						
PLACRS	FCA8	4621	4652	5231	-5300	5521			
PLOT	0050	-1653							
PLUS	ECFC	2532	-2534						
PLYARG	05E0	-647	648						
PLYEVL	DD40	-600							
PMBASE	D407	-756							
PNMI	E7B4	-1383	1605	1606					
PNMI1	E7BC	1384	-1386						
PNMIH	00E7	1268	-1605	1606					
PNMIL	00B4	1270	-1606						
POINT	0025	-102							
POKEY	D200	-664	665	666	667	668	669	670	671
		672	673	674	675	676	677	678	679
		680	681	682	683	684	685	686	687
		688	689	690	691	692	693		
POKMSK	0010	-184	1300	1308	1312	1318	1326	1333	1340
		1347	1353	2017	2019	2051	2053	2324	2350
		2352	2393	2394	4093	4094			
POKTAB	EDD2	2535	2538	-2707					
PORTA	D300	-766	1287	1364	1494	1501			
PORTB	D301	-767	1288	1368					
POT0	D200	-665	1511						
POT1	D201	-666							
POT2	D202	-667							
POT3	D203	-668							
POT4	D204	-669	1513						
POT5	D205	-670							
POT6	D206	-671							
POT7	D207	-672							
POTGO	D20B	-676	1517						
PRINTR	0050	-122							
PRINTV	E430	-54	3488	3826					
PRIOR	D01B	-723	1442						
PRMODE	EF1A	2988	3018	-3074					
PRNBUF	03C0	-536	2945	2991	3001				
PRNORG	EE78	-25	2881	2930					
PRNSPR	0014	-3102							
PRVOPN	0081	-138	883						
PSIOC	EF01	3048	-3050						
PTEMP	001F	-198	2987	2990					
PTIMOT	001C	-195	2938	3055	3065				

```
PTRIG0     027C          -388  1528
PTRIG1     027D          -389  1525
PTRIG2     027E          -390
PTRIG3     027F          -391
PTRIG4     0280          -392
PTRIG5     0281          -393
PTRIG6     0282          -394
PTRIG7     0283          -395
PTRLP      E8D1         -1521  1532
PUTADR     EE6D          2833  2840 -2868
PUTBC      EE43          2832 -2837
PUTCAR     000B         -3389
PUTCHR     000B           -88   900
PUTCNT     EE21          2817 -2823
PUTDAT     0080         -2758  2815
PUTDTO     EE01          2807 -2809
PUTLIN     F385          3712 -3936
PUTMSC     FCF3          4545  4760 -5355
PUTREC     0009           -87
PUTSEC     0050         -2753
PUTTXT     0009         -3387  3941
PWRONA     F3E4          4027  4040  4054 -4064
PWRUP      F125          3470  3474  3543 -3553  3663
PWRUP1     F128          3545 -3555
RADFLG     00FB          -635
RADON      0000          -636
RAMLO      0004          -171  3566  3567  3568  3572  3573  3773  3777
                         3780  3781  3782  3783  3786  3875  3877  3884
                         3888  3890  3891  3893  3914  3917  3918
RAMSIZ     02E4          -454  3632  3640  3815
RAMTOP     006A          -274  4068  4111  4122  4185  4207  4553  5144
                         5158
RANDOM     D20A          -675
RANGE      FA96          4268  4287  4933  4935 -4937  5500  5512
RANGE1     FAB7          4947 -4953
RANGE2     FABB          4952 -4955
RANGE3     FA9E          4939 -4941
RBLOK      EFE9          3134  3227 -3232
RBLOKV     E47A           -76  3133  3955
RCI1       E5A7           977   982  -987
RCI11      E5BF         -1008
RCI1A      E574           950  -957
RCI1B      E571          -954   958
RCI2       E5AC          -993
RCI3       E587           963  -969   988
RCI4       E5C3           971   984   995  1000 -1012
RCI6       E5B2          -998  1005
RDBAD      EE51         -2844  2851
RDBYTE     F310         -3870  3873
RDONLY     0087          -144  1022
READ       0052         -1642  3956
RECEIV     EAE2          1859  1910 -2068  2278
RECVDN     0039          -229  2075  2090  2142
```

```
SEND      EA6B        -1959  2246  2464
SENDDS    EC63         1880  1983 -2391  2641
SENDEN    EBF6         1711  1962  2228 -2305
SENDEV    E468          -70  1710  3202
SENDIN    EC8E         1798  1826 -2457
SERIN     D20D         -677  2134  2151
SERDUT    D20D         -691  1971  2012  2030
SETBSZ    EF34         3079  3083 -3087
SETDCB    EEE6         2966  3009 -3039
SETTAB    F82D        -4599  5663
SETVB1    E93D         1580 -1591  1592  1593
SETVBL    E912         1238 -1568
SETVBV    E45C          -66  1237  2670  3175  3208
SETVBX    EDBD         2236  2263  2274  2468 -2661
SEX       0000        -3392  3652  3937
SFH       EF64        -3166
SHFAMT    006F         -278  4276  4325  4821
SHFLOK    02BE         -431  4070  4471  4476  4481  4498
SHIFT1    F5B1         4277 -4280
SHIFT2    F610         4326 -4329
SHIFTD    F5AA        -4276  4279
SHIFTU    F608        -4325  4328
SIDWAY    0053        -1650
SIGNON    F223         3462 -3710
SIN       BD81         -611
SIO       E959         1705 -1745
SIOINT    E944         1708 -1725
SIOINV    E465          -69  1707  3829
SIOORG    E944          -23  1610  1721
SIOSB     F095         3233 -3342  3375
SIOSPR    0014        -2735
SIOV      E459          -65  1704  2827  2967  3010  3365
SIRHI     00EB        -2444  2445
SIRLO     0011        -2445
SIZEM     D00C         -708
SIZEP0    D008         -704
SIZEP1    D009         -705
SIZEP2    D00A         -706
SIZEP3    D00B         -707
SKCTL     D20F         -693  1735  2321  2345
SKRES     D20A         -690  2117  2347
SKSTAT    D20F         -679  1471  1481  2116  2577  2597  2620  2622
SOUNDR    0041         -239  1734  2361
SPACE     0020        -2904  3000
SPECIA    EF4B         3123 -3149
SPECIL    000E          -91   860   862
SPECL     F23F         3559 -3758
SQR       BEB1         -614
SRETRN    EB36         2136 -2141  2177
SRSTA     0040        -3109  3358
SRTIM2    0006        -1236  1487
SRTIMR    022B         -334  1479  1484  1488  1538  5772
SRTIRO    EB9F         2232 -2235
```

```
SRTIR1      EBC5          2259 -2262
SRTIR2      EBED          2281 -2286
SSFLAG      02FF          -476  1357  4304  5761  5763
SSKCTL      0232          -341  1733  2306  2320  2342  2344  2621
STACK     S 0000             0
STACKP      0318          -510  1746  2649
STATC       0053         -2754  2816  2831  2963  3047  3849
STATIS      000D           -90
STATU       F028          3123 -3271
STATUS      0030          -220  1865  1868  1883  1913  1930  1934  1936
                          1946  1960  2080  2094  2123  2129  2139  2647
STATVH      0002         -2744  2745  2820
STATVL      00EA         -2745  2818
STICK0      0278          -384  1499  1503  1521
STICK1      0279          -385
STICK2      027A          -386
STICK3      027B          -387
STIMER      D209          -689
STLOOP      E89C         -1494  1506
STORE       F917          4168  4182  4188  4190  4192  4200  4210  4212
                          4214  4216  4220  4222  4225  4227  4229  4235
                          4245  4247 -4723
STORE1      F91D         -4726
STRBEG      FC5C          4523 -5239
STRERR      F942          4742 -4746
STRIG0      0284          -396  1510
STRIG1      0285          -397
STRIG2      0286          -398
STRIG3      0287          -399
STRL        E8B5         -1509  1516
STROK       F946          4724  4732  4741  4745 -4748
STTMOT      EC79          1834  2273 -2417
SUBBFL      E677          1012  1073 -1154
SUBEND      FA7A         -4917  5471  5499
SUBTMP      029E          -417  4730  4735
SUCCES      0001          -135   912  1869  1914  1959  2079  2998  3021
                          3217  3230  3257  3271  3280  4090  4348  4959
SUSUAL      EB3A         -2144  2166  2169  2183
SV1H        00E9          1582 -1592  1593
SV1L        003D          1584 -1593
SV7H        00E8         -1535  1536
SV7L        0098         -1536
SWAP        FCB3          4362  4376  4396  4402  4407  4414  4426 -5311
SWAP1       FCC2         -5317  5324
SWAP3       FCD7          5315 -5328
SWAPA       FCB9          4973  5311 -5313
SWPFLG      007B          -286  4099  4885  4971  5199  5325  5327  5344
SWSTA       0080         -3110  3361
SYIRQ       E6F6          1255 -1294
SYIRQ1      E70B          1297 -1303
SYIRQ2      E71F          1305 -1311
SYIRQ3      E737          1313  1315 -1321
SYIRQ4      E74A          1323 -1329
```

```
SYIRQ5    E75A          1330 -1336
SYIRQ6    E767          1337 -1343
SYIRQ7    E779          1344 -1350
SYIRQ8    E793          1350 -1362
SYIRQ9    E79E          1363 -1366
SYIRQA    E7A9          1367 -1370
SYIRQB    E7B2          1247  1248  1249  1252  1253  1254  1373 -1375
SYRTI     E7B3          1246 -1376
SYSVB1    E7DD          1405  1408 -1410
SYSVB2    E7F9          1421 -1423
SYSVB3    E808          1428 -1430
SYSVB4    E857          1458 -1460
SYSVB6    E8E8          1483 -1537
SYSVB7    E898          1480  1485 -1492  1535  1536  1539
SYSVBA    E869          1465 -1468
SYSVBB    E859         -1461  1469
SYSVBL    E7D1          1239  1259 -1404
SYSVBV    E45F           -67
SYVB6A    E87C          1473  1476 -1479
TAB       F810         -4587  4597  5653
TAB1      F823          4590  4593 -4595
TAB2      F82A          4594 -4598
TABMAP    02A3           -422  4103  5019  5021  5027  5028  5036
TBLENT    F0E3         -3487  3515  3618
TBLLEN    000E         -3515  3617
TDHI      00EA         -2448  2449
TDLO      00D1         -2449
TEMP      023E           -357  1685  1686  1920
TEMP1     0312           -505  2499  2510
TEMP2     0314           -506
TEMP3     0315           -507  2587  2605
TEMPHI    0002         -1685  1686  1903
TEMPLO    003E         -1686  1899
TIMER1    030C           -500  2493  2495  2498  2502  2583  2584
TIMER2    0310           -503  2489  2490  2492  2496  2500
TIMFLG    0317           -509  2088  2241  2268  2295  2573  2592  2667
TIMIT     EBA9         -2241  2242
TIMIT1    EBCF         -2268  2269
TIMOUT    008A           -147  1937  2093
TINDEX    0293           -412  4097
TMPCHR    0050           -255  4330  4334  5075  5078
TMPCOL    02B9           -427
TMPLBT    02A1           -420  4796  4805  4809
TMPROW    02B8           -426  5292  5301
TMPX1     029C           -415
TOADR     0066         -4012  4663  4667  5074  5077  5087  5089  5091
TONE1     0002         -3117  3193
TONE2     0001         -3118  3166
TOUT      EB0C          2089 -2093  2595
TOUT1     ED48          2575 -2594
TRAMSZ    0006           -172  3574  3631  3645  3674  3677  3692  3775
                        3787  3790  3814
TRIGO     D010           -744  1509
```

```
TRIG1     D011          -745
TRIG2     D012          -746
TRIG3     D013          -747
TRNRCD    0089          -146  1008
TSTAT     0319          -511  1864  1947
TSTCT1    FC8F         -5277  5283
TSTCT2    FC9C          5279 -5284
TSTCTL    FC8D          4410  4502 -5276
TSTDAT    0007          -173  3630  3637  3675  3680  3698  3778  3785
TWICE     EE4F         -2842  2846
TXTCOL    0291          -411  4261
TXTMSC    0294          -413  4113  4115
TXTOLD    0296          -414
TXTROW    0290          -410  4259  5319  5322
UNLOCK    0024          -101
UPDNCM    F787         -4522  4529
USAREA    0480          -551
VBATRA    E7EB          1413 -1417
VBREAK    0206          -315  1374
VBWAIT    F496         -4153  4155
VCOUNT    D40B          -759  2581  2607  4153
VCTABL    E480           -20  1244  1713  3607  4059
VDELAY    D01C          -724
VDSLST    0200          -312  1244  1385
VECTBL    E400           -19
VIMIRQ    0216          -323  1293
VINTER    0204          -314  1369
VKEYBD    0208          -316  1349  4059
VPRCED    0202          -313  1365
VSCROL    D405          -755
VSERIN    020A          -317  1310  1713
VSEROC    020E          -319  1320
VSEROR    020C          -318  1302
VTIMR1    0210          -320  1328
VTIMR2    0212          -321  1335
VTIMR4    0214          -322
VVBLKD    0224          -330  1534
VVBLKI    0222          -329  1396  1590
WAIT      EA1A         -1895  2470
WAITER    EC9F          1839 -2468
WAITTM    EF7C         -3176  3177
WARMST    0008          -176  3555  3561  3565  3840  3967
WARMSV    E474           -74  1390  3464
WATCOM    E9D7          1814 -1834
WCI1      E605          1048  1053 -1058
WCI1A     E5D4          1019 -1026
WCI1B     E5D1         -1023  1027
WCI2      E60A         -1064
WCI3      E5E5          1032 -1038  1059
WCI4      E5EB          1035 -1041
WCI5      E615          1042  1055  1066 -1073
WDLR      EFC6         -3211  3214
WFAK      F087          3323 -3331
```

```
WFAK1       F08C          3331 -3334
WFL         F060         -3310  3321
WIRGHI      0000         -1674  2235
WIRGLO      00B4         -1673  2234
WMODE       0289          -402  3164  3188  3192  3277
WOK         EA3D         -1920
WRITE       0057         -1643  2813
WRITEC      0057         -2903  3074
WRONLY      0083          -140   953
WSIOSB      F0D2          3263  3288  3296 -3370
WSIRG       000F         -1677  2230
WSYNC       D40A          -758  5335
WTLR        F046          3285 -3290
XBOOT       F361          3904 -3906
XITVBL      E93E          1240  1260  1429 -1597
XITVBV      E462           -68
XMTDON      003A          -230  1967  1980  2049
XXIT        E805          1424 -1429
ZERIT       EC71         -2399  2402
ZERORM      F138         -3564
ZIOCB       0020          -200
ZOSRAM      F160          3562 -3587
ZOSRM2      F163         -3589  3592
ZOSRM3      F16E         -3594  3596
ZTBUF       F04A         -3292  3294
ZTEMP1      00F5          -631
ZTEMP3      00F9          -633
ZTEMP4      00F7          -632
```