1571

Source

DoS

&

Controller

APPLICATION		REVISIONS			
EXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED
	1571				

MODORE PART #	STATUS				
315081-01	ACTIVE				

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES

TOLERANCES
ANGLES ±1°
2 PLACE DECIMALS ±.02
3 PLACE DECIMALS ±.010

DRWN	
SYSTEMS ENG	
TEST ENG	
CIRCUIT ENG	
COMP ENG	

COMMODORE

TITLE

DISK OPERATING SYSTEM AND
CONTROLLER ROUTINES

SIZE	DRAWING NO.
A	315081

SCALE	SHEET 1 OF 576

```
****   ******************************
*       B E T A 7 - 05/29/85         *
************************************
* 1) FILL MEMORY WITH $FF            *
* 2) DOWNLOAD CODE                   *
* 3) CHANGE THE FOLLOWING:           *
*          $92=$8000                 *
*          $25=$8001                 *
*          $42=$C000                 *
************************************
```

```
Error Addr  Code          Seq   Source statement

                            1          .include  com
                            2   ;*********************************************
                            3   ;*                                           *
                            4   ;*   COMMODORE BUSINESS MACHINES SOFTWARE     *
                            5   ;*                                           *
                            6   ;*********************************************
                            7   ;*                                           *
                            8   ;*    11     55555    77777     11           *
                            9   ;*     1     5            7      1           *
                           10   ;*     1     5555       7        1           *
                           11   ;*     1         5      7        1           *
                           12   ;*   11111   5555       7      11111         *
                           13   ;*                                           *
                           14   ;*    FFFFFF   AAAAAA   SSSSSS   TTTTTT      *
                           15   ;*    F        A    A   S          T         *
                           16   ;*    FFFF     AAAAAA   SSSSSS     T         *
                           17   ;*    F        A    A        S     T         *
                           18   ;*    F        A    A   SSSSSS     T         *
                           19   ;*                                           *
                           20   ;*  SSSSS  EEEEE  RRRRR  IIIII  AAAAA  L     *
                           21   ;*  S      E      R   R    I    A   A  L     *
                           22   ;*  SSSSS  EEEE   RRR      I    AAAAA  L     *
                           23   ;*      S  E      R  R     I    A   A  L     *
                           24   ;*  SSSSS  EEEEE  R   R  IIIII. A   A  LLLLL *
                           25   ;*                                           *
                           26   ;*  DDDD   IIIII  SSSSS  K    K             *
                           27   ;*  D   D    I    S      K   K              *
                           28   ;*  D   D    I    SSSSS  KKK                *
                           29   ;*  D   D    I        S  K   K              *
                           30   ;*  DDDD   IIIII  SSSSS  K    K             *
                           31   ;*                                           *
                           32   ;*                                           *
                           33   ;*********************************************
                           34   ;*                                           *
                           35   ;*   DISK OPERATING SYSTEM AND CONTROLLER     *
                           36   ;*   ROUTINES.                               *
                           37   ;*                                           *
                           38   ;*   COPYRIGHT (C) 1985 BY                   *
                           39   ;*   COMMODORE BUSINESS MACHINES (CBM)       *
                           40   ;*                                           *
                           41   ;*                                      DS   *
                           42   ;*                                           *
                           43   ;*********************************************
                           44   ;
                           45   ;*********************************************
                           46   ;*   THIS SOFTWARE IS FURNISHED FOR USE IN   *
                           47   ;*  THE 1571 FAST SERIAL FLOPPY DISK DRIVE.  *
                           48   ;*                                           *
                           49   ;*   COPIES THEREOF MAY NOT BE PROVIDED OR   *
                           50   ;*  MADE AVAILABLE FOR USE ON ANY OTHER      *
```

```
Error Addr  Code          Seq   Source statement

                           51   ;*  SYSTEM.                                  *
                           52   ;*                                           *
                           53   ;*   THE INFORMATION IN THIS DOCUMENT IS      *
                           54   ;*  SUBJECT TO CHANGE WITHOUT NOTICE.         *
                           55   ;*                                           *
                           56   ;*   NO RESPONSIBILITY IS ASSUMED FOR        *
                           57   ;*  RELIABILITY OF THIS SOFTWARE.            *
                           58   ;*                                           *
                           59   ;*********************************************
                           60           .include  wd1770
                           61           .subttl 'wd1770.src'
```

```
Error Addr  Code          Seq   Source statement

                           63   wdtest  .macro
                           64           .ife <*!.$03     ; lower two bits cannot be zero
                           65           nop              ; fill address error
                           66           .endif
                           67           .endm
                           68
                           69   nodrrd  .macro           ; read nodrv,x absolute
                           70           .byte $bd,$ff,$00
                           71           .endm
                           72
                           73   nodrwr  .macro           ; write nodrv,x absolute
                           74           .byte $9d,$ff,$00
                           75           .endm
                           76           .include  var
```

```
Error Addr  Code           Seq   Source statement

                            78           .subttl 'var.src'
                            79
            =0000           80   *=0
                            81
      0000  =0006           82   jobs     ***+6          ; job que
                            83
                            84   ; controller job types
                            85
            =0080           86           read     =$80
            =0088           87           fread    =read+8
            =0090           88           write    =$90
            =00A0           89           wverfy   =$a0
            =00B0           90           seek     =$b0
            =00B8           91           secsek   =seek+8
            =00C0           92           bump     =$c0
            =00D0           93           jumpc    =$d0
            =00E0           94           exec     =$e0
            =00F0           95           frmt     =$f0
                            96
                            97   ;job returns
                            98
                            99           ;01 = ok
                           100           ;02 = can't find sector
                           101           ;03 = no sync
                           102           ;04 = not a data block
                           103           ;05 = data block checksum
                           104           ;06 = format error
                           105           ;07 = verify error
                           106           ;08 = write protect error
                           107           ;09 = header block checksum
                           108           ;10 = data extends into next block
                           109           ;11 = disk id mismatch
                           110
      0006  =0012          111   hdrs     ***+12         ; job headers
      0012  =0016          112   dskid    ***+4          ; master copy of disk id
                           113
      0016  =001B          114   header   ***+5          ; image of last header
                           115
      001B                 116   info                    ; **** 1571 ****
      001B                 117   fmhdpt                  ; **** 1571 ****
      001B  =001C          118   actjob   ***+1          ; controller's active job
      001C  =001E          119   wpsw     ***+2          ; write protect change flag
      001E  =0020          120   lwpt     ***+2          ; last state of wp switch
      0020  =0022          121   drvst    ***+2
                           122                           ;$10 = time out state
                           123                           ;$20 = motor just running
                           124                           ;$40 = stepping
                           125                           ;$80 = accelerating
                           126                           ;$00 = no drive active
                           127
```

```
Error Addr  Code          Seq   Source statement

     0022   =0023         128   drvtrk    *=*+1          ; current track
            =0024         129             *=*+1          ; slowd flag
                          130
     0024                 131   mfmhdr                   ; **** 1571 ****
     0024   =002E         132   stab      *=*+10
                          133
                          134   ;    pointers
                          135
     002E   =0030         136   savpnt    *=*+2
     0030   =0032         137   bufpnt    *=*+2
     0032   =0034         138   hdrpnt    *=*+2
                          139
     0034   =0035         140   gcrpnt    *=*+1
                          141
     0035                 142   acltim2                  ;  **** 1571 ****
     0035   =0036         143   gcrerr    *=*+1          ;  indicates gcr decode error
     0036   =0037         144   bytcnt    *=*+1
                          145
     0037                 146   fastsr                   ;  **** 1571 ****
     0037   =0038         147   bitcnt    *=*+1
     0038   =0039         148   bid       *=*+1
     0039   =003A         149   hbid      *=*+1
     003A   =003B         150   chksum    *=*+1
                          151
     003B                 152   tsttrk                   ;  **** 1571 ****
     003B                 153   switch                   ;  **** 1571 ****
     003B   =003C         154   hinib     *=*+1
                          155
     003C                 156   cpmit                    ;  **** 1571 ****
     003C   =003D         157   byte      *=*+1
     003D   =003E         158   drive     *=*+1
     003E   =003F         159   cdrive    *=*+1
     003F   =0040         160   jobn      *=*+1
     0040   =0041         161   tracc     *=*+1
     0041   =0042         162   nxtjob    *=*+1
     0042   =0043         163   nxtrk     *=*+1
     0043   =0044         164   sectr     *=*+1
                          165
     0044                 166   mfmsiz_hi                ;  **** 1571 ****
     0044   =0045         167   work      *=*+1
     0045   =0046         168   job       *=*+1
                          169
     0046                 170   ctl_dat                  ;  **** 1571 ****
     0046   =0047         171   ctrack    *=*+1
     0047   =0048         172   dbid      *=*+1          ;  data block id
     0048   =0049         173   acltim    *=*+1          ;  acel time delay
     0049   =004A         174   savsp     *=*+1          ;  save stack pointer
     004A   =004B         175   steps     *=*+1          ;  steps to desired track
     004B   =004C         176   tmp       *=*+1
     004C   =004D         177   csect     *=*+1
```

```
Error Addr  Code           Seq   Source statement

     004D   =004E          178   nexts     ***+1
     004E   =004F          179   nxtbf     ***+1          ;  pointer at next gcr source buffer
     004F   =0050          180   nxtpnt    ***+1          ;  and next gcr byte location in buffer
     0050   =0051          181   gcrflg    ***+1          ;  buffer in gcr image
     0051   =0052          182   ftnum     *=*+1          ;  current format track
     0052   =0056          183   btab      ***+4
     0056   =005E          184   gtab      ***+8
                           185
     005E                  186   dkmode                   ;  **** 1571 ****
     005E   =005F          187   as        *=*+1          ;  # of steps to acel
                           188
     005F                  189   ctl_cmd                  ;  **** 1571 ****
     005F   =0060          190   af        *=*+1          ;  acel. factor
                           191
     0060                  192   minsek                   ;  **** 1571 ****
     0060   =0061          193   aclstp    *=*+1          ;  steps to go
                           194
     0061                  195   maxsek                   ;  **** 1571 ****
     0061   =0062          196   rsteps    *=*+1          ;  # of run steps
     0062   =0064          197   nxtst     *=*+2          ;  1541 pointer
                           198
     0064                  199   cur_trk                  ;  **** 1571 ****
     0064   =0065          200   minstp    *=*+1          ;  min reqired to acel
                           201
                           202
                           203   ;  constants
                           204
            =0100          205   ovrbuf    =$0100         ;  top of stack
            =0006          206   numjob    =6             ;  number of jobs
            =0050          207   jmpc      =$50           ;  jump command
            =0040          208   bumpc     =$40           ;  bump command
            =0060          209   execd     =$60           ;  execute command
            =0300          210   bufs      =$0300         ;  start of buffers
            =0300          211   buff0     =bufs
            =0400          212   buff1     =bufs+$100
            =0500          213   buff2     =bufs+$200
            =0002          214   tolong    =$2            ;  format errors
            =0003          215   tomany    =$3
            =0004          216   tobig     =$4
            =0005          217   tosmal    =$5
            =0006          218   notfnd    =$6
            =0024          219   skip1     =$24           ;  bit zp
            =002C          220   skip2     =$2c           ;  bit abs
            =0045          221   toprd     =69            ;  top of read overflo buffer on a read
            =0045          222   topwrt    =69            ;  top of write overflo buffer on a write
            =0005          223   numsyn    =5             ;  gcr byte count for size of sync area
            =000B          224   gap1      =11            ;  gap after header to  clear erase in gcr bytes
            =0004          225   gap2      =4             ;  gap after block min size
            =0006          226   rdmax     =6             ;  sector distance wait
            =0009          227   wrtmin    =9
```

```
Error Addr  Code           Seq   Source statement

            =000C          228   wrtmax    =12
            =0020          229   tim       =32             ;  irq rate for 8 ms @ 1 Mhz
            =0040          230   tim2      =64             ;  irq rate for 8 ms @ 2 Mhz
            =1C00          231   byt_clr   =$1C00
            =00FF          232   mscnt     =255
```

```
Error Addr  Code           Seq    Source statement

                           234
                           235    ;*******************************
                           236    ;*  equates
                           237    ;*******************************
                           238
            =0065          239    zp2       =*              ; save zero page for dos
            =C000          240    rom       =$c000          ; first rom address
                           241
            =0080          242    lrf       =$80            ; last record flag
            =0040          243    dyfile    =$40            ; dirty flag for rr file
            =0020          244    ovrflo    =$20            ; rr print overflow
            =0006          245    nssl      =6              ; # of side-sector links
            =0010          246    ssioff    =4+nssl+nssl    ; offset into ss for data block ptrs
            =0078          247    nssp      =120            ; # of ptrs in ss
            =0006          248    mxchns    =6              ; max # channels in system
            =0012          249    maxsa     =18             ; max sa # +1
            =0007          250    vererr    =7              ; controller verify error
            =000D          251    cr        =$0d            ; carriage return
            =0005          252    bfcnt     =5              ; available  buffer count
            =000A          253    cbptr     =bfcnt+bfcnt    ; command buffer ptr
            =0005          254    errchn    =mxchns-1       ; error channel #
            =0010          255    errsa     =16             ; error channel sa #
            =0004          256    cmdchn    =mxchns-2       ; command channel #
            =000F          257    lxint     =%00001111      ; power up linuse (logical index usage
            =0006          258    blindx    =6              ; bam lindx for floating bams
            =000F          259    cmdsa     =15             ; command channel sa #
            =0002          260    apmode    =2              ; open append mode
            =0003          261    mdmode    =3              ; open modify mode
            =0000          262    rdmode    =0              ; open read mode
            =0001          263    wtmode    =1              ; open write mode
            =0004          264    reltyp    =4              ; open relative type
            =0007          265    dirtyp    =7              ; open direct type
            =0001          266    seqtyp    =1              ; open sequential type
            =0002          267    prgtyp    =2              ; open program type
            =0003          268    usrtyp    =3              ; open user type
            =0007          269    typmsk    =7              ; mask for type bits
            =0011          270    irsa      =17             ; internal read sa #
            =0012          271    iwsa      =18             ; internal write sa #
            =0002          272    dosver    =2              ; dos version
            =0042          273    fm2030    =$42            ; 2030 format version
            =0041          274    fm4040    =$41            ; 4040 format version
            =0005          275    mxfils    =5              ; max # filenames in string
            =0018          276    dirlen    =24             ; directory length used
            =001B          277    nbsiz     =27             ; nambuf text size
            =0029          278    cmdlen    =41             ; length of command buffer
            =0001          279    drvmsk    =1              ; drv 0 or 1
```

```
Error Addr  Code           Seq   Source statement

                           281
                           282   ;*****************************
                           283   ;*      i/o definitions      *
                           284   ;*****************************
                           285   ;
           =003F           286   unlsn    =$3f            ; ieee unlisten command
           =005F           287   untlk    =$5f            ; ieee untalk command
           =0000           288   notrdy   =$0             ; not ready
           =0080           289   talker   =$80            ; ieee talker flag
           =0001           290   lisner   =1              ; ieee listener flag
           =0080           291   eoiout   =$80            ; talk with eoi
           =0008           292   eoisnd   =$08            ; not(eoi) to send
           =0008           293   eoi      =$08            ; not(eoi) to send
           =0088           294   rdytlk   =$88            ; talk no eoi
           =0001           295   rdylst   =$1             ; ready to listen
           =0089           296   rndrdy   =rdytlk+rdylst ; random chnrdy
           =0081           297   rndeoi   =eoiout+rdylst ; random w/ eoi
                           298
                           299   ;*********************************************************************
                           300   ;i/o registers
                           301
                           302   ; mos 6522-a
                           303
           =1800           304           *  =$1800
                           305   ;serial ieee control port
                           306
     1800  =1801           307   pb       *=*+1           ; serial port
                           308                           ; bit 0: data in
                           309                           ; bit 1: data out
                           310                           ; bit 2: clk in
                           311                           ; bit 3: clk out
                           312                           ; bit 4: atn ack
                           313                           ; bit 5: dev jumper
                           314                           ; bit 6: dev jumper
                           315                           ; bit 7: atn in
     1801  =1802           316   pa1      *=*+1
                           317
     1802  =1803           318   ddrb1    *=*+1           ; serial data dir
     1803  =1804           319   ddra1    *=*+1           ; unused ddra
     1804  =1805           320   t1lc1    *=*+1           ; timer 1 low counter
     1805  =1806           321   t1hc1    *=*+1           ; timer 1 hi counter
     1806  =1807           322   t1ll1    *=*+1           ; timer 1 low latch
     1807  =1808           323   t1hl1    *=*+1           ; timer 1 hi latch
     1808  =1809           324   t2lc1    *=*+1           ; timer 2 low counter
     1809  =180A           325   t2hc1    *=*+1           ; timer 2 hi counter
     180A  =180B           326   sr1      *=*+1           ; shift reg
     180B  =180C           327   acr1     *=*+1           ; aux control reg
     180C  =180D           328   pcr1     *=*+1
     180D  =180E           329   ifr1     *=*+1
     180E  =180F           330   ier1     *=*+1
```

'var.src' VAR.SRC

```
Error Addr  Code          Seq   Source statement

     180F   =1810          331   potA1     ***+1           ; bit 0: track 0 sense, 1=trk0
                           332                             ; bit 1: fast serial direction, 1=out
                           333                             ; bit 2: side select, 1=side 1
                           334                             ; bit 3: ------------
                           335                             ; bit 4: ------------
                           336                             ; bit 5: 1/2 Mhz, 1=2 Mhz
                           337                             ; bit 6: atn out
                           338                             ; bit 7: byte ready, 0=byte
                           339
                           340   ;*****************************************************************
                           341
                           342   ;  bits for serial handshake
                           343
           =0001           344   datin     =$1             ; data in line
           =0002           345   datout    =$2             ; data out
           =0004           346   clkin     =$4             ; clock in
           =0008           347   clkout    =$8             ; clock out
           =0010           348   atna      =$10            ; atna ack
           =0080           349   atn       =$80            ; atn in
                           350
           =0008           351   led0      =8              ; act led
           =0000           352   led1      =0              ; no led
           =1C00           353   ledprt    =$1c00          ; on  pb  of  $1c00
           =1C02           354   ledout    =$1c02          ; ddrb  of  $1c00  for  output-led
                           355
                           356   ;*****************************************************************
                           357
           =2000           358   *=$2000
                           359   ; western digital floppy disk cntrl
                           360
                           361   ; ******* BEWARE THIRD PARTY AND INHOUSE VENDORS THIS PART WILL GO AWAY ***
                           362
     2000  =2001           363   wdstat    ***+1           ; command/status reg
     2001  =2002           364   wdtrk     ***+1           ; track reg
     2002  =2003           365   wdsec     ***+1           ; sector reg
     2003  =2004           366   wddat     ***+1           ; data reg
                           367
                           368   ;*****************************************************************
                           369
           =4000           370   *=$4000
                           371   ; 6526a serial chip (about it)... shift registers in the 6522 suck...
                           372
                           373   ; ******* BEWARE THIRD PARTY AND INHOUSE VENDORS THIS PART WILL GO AWAY ***
                           374
           =4001           375           ***+1           ; unused port a
           =4002           376           ***+1           ; unused port b
           =4003           377           ***+1           ; unused dd port a
           =4004           378           ***+1           ; unused dd port b
     4004  =4005           379   tima_l    ***+1           ; timer a used for the baud rate generator
     4005  =4006           380   tima_h    ***+1           ; *
```

```
Error Addr  Code          Seq   Source statement

            =4007         381           ***+1           ; unused timb_l
            =4008         382           ***+1           ; unused timb_h
            =400C         383           ***+4           ; unused tod
     400C   =400D         384   sdr     ***+1           ; serial data register
     400D   =400E         385   icr     ***+1           ; interrupt control register
     400E   =400F         386   cra     ***+1           ; control register a
            =4010         387           ***+1           ; unused
                          388
                          389   ;*****************************************************************************
```

```
Error Addr  Code          Seq   Source statement

                          391
                          392   ;permanent address variables
                          393
            =0065         394   *=zp2
                          395
      0065  =0067         396   vnmi      ***+2           ; indirect for nmi
                          397
      0067                398   cmd_trk                   ; **** 1571 ****
      0067  =0068         399   nmiflg    ***+1
      0068  =0069         400   autofg    ***+1
      0069  =006A         401   secinc    ***+1           ; sector inc for seq
      006A  =006B         402   revcnt    ***+1           ; error recovery count
                          403
            =0300         404   bufs      =$300           ; start of data bufs
            =0300         405   fbufs     =bufs           ; format download image
                          406
                          407   ;*
                          408   ;*********************************
                          409   ;*
                          410   ;*       zero page variables
                          411   ;*
                          412   ;*********************************
                          413   ;*
      006B  =006D         414   usrjmp    ***+2           ;  user jmp table ptr
      006D  =006F         415   bmpnt     ***+2           ;  bit map pointer
      006F  =0075         416   temp      ***+6           ;  temp work space
      0075  =0077         417   ip        ***+2           ;  indirect ptr variable
      0077  =0078         418   lsnadr    ***+1           ;  listen address
      0078  =0079         419   tlkadr    ***+1           ;  talker address
      0079  =007A         420   lsnact    ***+1           ;  active listener flag
      007A  =007B         421   tlkact    ***+1           ;  active talker flag
      007B  =007C         422   adrsed    ***+1           ;  addressed flag
      007C  =007D         423   atnpnd    ***+1           ;  attention pending flag
      007D  =007E         424   atnmod    ***+1           ;  in atn mode
      007E  =007F         425   prgtrk    ***+1           ;  last prog accessed
      007F  =0080         426   drvnum    ***+1           ;  current drive #
      0080  =0081         427   track     ***+1           ;  current track
      0081  =0082         428   sector    ***+1           ;  current sector
      0082  =0083         429   lindx     ***+1           ;  logical index
      0083  =0084         430   sa        ***+1           ;  secondary address
      0084  =0085         431   orgsa     ***+1           ;  original sa
      0085  =0086         432   data      ***+1           ;  temp data byte
                          433
            =006F         434   t0        =temp
            =0070         435   t1        =temp+1
            =0071         436   t2        =temp+2
            =0072         437   t3        =temp+3
            =0073         438   t4        =temp+4
      0086  =0087         439   r0        ***+1
      0087  =0088         440   r1        ***+1
```

```
Error Addr  Code          Seq   Source statement

     0088   =0089          441   r2        *=*+1
     0089   =008A          442   r3        *=*+1
     008A   =008B          443   r4        *=*+1
     008B   =008F          444   result    *=*+4
     008F   =0094          445   accum     *=*+5
     0094   =0096          446   dirbuf    *=*+2
                          447
     0096                 448   cmd_sec                   ;  **** 1571 ****
     0096   =0097          449   icmd      *=*+1           ;  ieee cmd in
                          450
     0097                 451   cpmsek                    ;  **** 1571 ****
     0097   =0098          452   mypa      *=*+1           ;  my pa flag
     0098   =0099          453   cont      *=*+1           ;  bit counter for ser
                          454   ;*
                          455   ;**********************
                          456   ;*
                          457   ;*  zero page array
                          458   ;*
                          459   ;**********************
                          460   ;*
     0099   =00A7          461   buftab    *=*+cbptr+4     ;  buffer byte pointers
            =00A3         462   cb=buftab+cbptr
     00A7   =00AE          463   buf0      *=*+mxchns+1
     00AE   =00B5          464   buf1      *=*+mxchns+1
     00B5                 465   nbkl
     00B5   =00BB          466   recl      *=*+mxchns
     00BB                 467   nbkh
     00BB   =00C1          468   rech      *=*+mxchns
     00C1   =00C7          469   nr        *=*+mxchns
     00C7   =00CD          470   rs        *=*+mxchns
     00CD   =00D3          471   ss        *=*+mxchns
     00D3   =00D4          472   fiptr     *=*+1           ;  file stream 1 pointer
                          473
     00D4   =00D5          474   recptr    *=*+1
     00D5   =00D6          475   ssnum     *=*+1
     00D6   =00D7          476   ssind     *=*+1
     00D7   =00D8          477   relptr    *=*+1
     00D8   =00DD          478   entsec    *=*+mxfils      ;  sector of directory entry
     00DD   =00E2          479   entind    *=*+mxfils      ;  index of directory entry
     00E2   =00E7          480   fildrv    *=*+mxfils      ;  default flag, drive #
     00E7   =00EC          481   pattyp    *=*+mxfils      ;  pattern,replace,closed-flags,type
     00EC   =00F2          482   filtyp    *=*+mxchns      ;  channel file type
     00F2   =00F8          483   chnrdy    *=*+mxchns      ;  channel status
     00F8   =00F9          484   eoiflg    *=*+1           ;  temp eoi
     00F9   =00FA          485   jobnum    *=*+1           ;  current job #
     00FA   =00FF          486   lrutbl    *=*+mxchns-1    ;  least recently used table
     00FF   =0101          487   nodrv     *=*+2           ;  no drive flag
                          488
     0101   =0103          489   dskver    *=*+2           ;  disk version from 18.0
            =0103         490   zpend=*
```

```
Error Addr  Code          Seq   Source statement

                          491                              ;  stack area
                          492
            =0146         493           *=$0100+$46        ;  ****** STACK ******
                          494
      0146  =01AF         495   bam_one ***+105            ;  **** 1571 vars ****
      01AF  =01B0         496   dside   ***+1              ;  side flag
      01B0  =01B1         497   mfmcmd  ***+1              ;  mfm controller command
      01B1  =01B2         498   side    ***+1              ;  side flag
      01B2  =01B3         499   fmtsid  ***+1
                          500
            =01BB         501           *=$01bb            ;  end of vars start overflow buffer
                          502
            =0200         503   *=$200
      0200  =022A         504   cmdbuf  ***+cmdlen+1
      022A  =022B         505   cmdnum  ***+1              ;  command #
      022B  =023E         506   lintab  ***+maxsa+1        ;  sa:lindx table
      023E  =0244         507   chndat  ***+mxchns         ;  channel data byte
      0244  =024A         508   lstchr  ***+mxchns         ;  channel last char ptr
      024A  =024B         509   type    ***+1              ;  active file type
      024B  =024C         510   strsiz  ***+1
      024C  =024D         511   tempsa  ***+1              ;  temporary sa
      024D  =024E         512   cmd     ***+1              ;  temp job command
                          513
      024E  =024F         514   lstsec  ***+1              ;
      024F  =0251         515   bufuse  ***+2              ;  buffer allocation
      0251  =0253         516   mdirty  ***+2              ;  bam 0 & 1 dirty flags
                          517
      0253  =0254         518   entfnd  ***+1              ;  dir-entry found flag
      0254  =0255         519   dirlst  ***+1              ;  dir listing flag
      0255  =0256         520   cmdwat  ***+1              ;  command waiting flag
      0256  =0257         521   linuse  ***+1              ;  lindx use word
      0257  =0258         522   lbused  ***+1              ;  last buffer used
      0258  =0259         523   rec     ***+1
      0259  =025A         524   trkss   ***+1
      025A  =025B         525   secss   ***+1
```

```
Error Addr  Code           Seq   Source statement

                           527   ;*
                           528   ;*********************************
                           529   ;*
                           530   ;*  ram array area
                           531   ;*
                           532   ;*********************************
                           533   ;*
     025B   =0260          534   lstjob    ***+bfcnt       ;  last job
     0260   =0266          535   dsec      ***+mxchns      ;  sector of directory entry
     0266   =026C          536   dind      ***+mxchns      ;  index of directory entry
     026C   =026D          537   erword    ***+1           ;  error word for recovery
     026D   =026E          538   erled     ***+1           ;  error led mask for flashing
     026E   =026F          539   prgdrv    ***+1           ;  last program drive
     026F   =0270          540   prgsec    ***+1           ;  last program sector
     0270   =0271          541   wlindx    ***+1           ;  write lindx
                           542
     0271                  543   mfmsiz_lo                 ;  **** 1571 ****
     0271   =0272          544   rlindx    ***+1           ;  read lindx
     0272   =0274          545   nbtemp    ***+2           ;  # blocks temp
                           546
     0274   =0275          547   cmdsiz    ***+1           ;  command string size
     0275   =0276          548   char      ***+1           ;  char under parser
     0276   =0277          549   limit     ***+1           ;  ptr limit in compar
     0277   =0278          550   f1cnt     ***+1           ;  file stream 1 count
     0278   =0279          551   f2cnt     ***+1           ;  file stream 2 count
     0279   =027A          552   f2ptr     ***+1           ;  file stream 2 pointer
                           553
                           554   ;  parser tables
     027A   =0280          555   filtbl    ***+mxfils+1    ;  filename pointer
     0280   =0285          556   filtrk    ***+mxfils      ;  1st link/track
     0285   =028A          557   filsec    ***+mxfils      ;           /sector
                           558
                           559   ;  channel tables
                           560
     028A   =028B          561   patflg    ***+1           ;  pattern presence flag
     028B   =028C          562   image     ***+1           ;  file stream image
     028C   =028D          563   drvcnt    ***+1           ;  number of drv searches
     028D   =028E          564   drvflg    ***+1           ;  drive search flag
     028E   =028F          565   lstdrv    ***+1           ;  last drive w/o error
     028F   =0290          566   found     ***+1           ;  found flag in dir searches
     0290   =0291          567   dirsec    ***+1           ;  directory sector
     0291   =0292          568   delsec    ***+1           ;  sector of 1st avail entry
     0292   =0293          569   delind    ***+1           ;  index  "
     0293   =0294          570   lstbuf    ***+1           ;  =0 if last block
     0294   =0295          571   index     ***+1           ;  current index in buffer
     0295   =0296          572   filcnt    ***+1           ;  counter, file entries
     0296   =0297          573   typflg    ***+1           ;  match by type flag
     0297   =0298          574   mode      ***+1           ;  active file mode (r,w)
     0298   =0299          575   jobrtn    ***+1           ;  job return flag
     0299   =029A          576   eptr      ***+1           ;  ptr for recovery
```

```
Error Addr  Code          Seq   Source statement

     029A   =029B         577   toff      *=*+1          ;  total track offset
     029B   =029D         578   ubam      *=*+2          ;  last bam update ptr
     029D   =02A1         579   tbam      *=*+4          ;  track # of bam image
                          580
                          581   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                           611            .subttl 'burst'
            =8000          612            *=$8000
                           613
     8000   FF             614    signature_lo .byte $FF  ; <<<< TO BE DETERMINED
     8001   FF             615    signature_hi .byte $FF  ; <<<< TO BE DETERMINED
                           616
                           617
     8002   53 2F 57       618            .byte  'S/W - DAVID G SIRACUSA',$OD,"H/W - GREG BERLIN",$OD,'1985',$OD
     8005   20 2D 20
     8008   44 41 56
     800B   49 44 20
     800E   47 20 53
     8011   49 52 41
     8014   43 55 53
     8017   41 OD 48
     801A   2F 57 20
     801D   2D 20 47
     8020   52 45 47
     8023   20 42 45
     8026   52 4C 49
     8029   4E OD 31
     802C   39 38 35
     802F   OD
                           619
     8030                  620    burst_routines
                           621
     8030   AD 0274        622            lda  cmdsiz     ; check command size
     8033   C9 03          623            cmp  #3
     8035   90 2E          624            bcc  realus
                           625
     8037   AD 0202        626            lda  cmdbuf+2   ; get command
     803A   85 3B          627            sta  switch     ; save info
     803C   29 1F          628            and  #$1f
     803E   AA             629            tax             ; command info
     803F   OA             630            asl  a
     8040   A8             631            tay
     8041   B9 808E        632            lda  cmdtbb,y
     8044   85 75          633            sta  ip
     8046   B9 808F        634            lda  cmdtbb+1,y
     8049   85 76          635            sta  ip+1
     804B   EO 1E          636            cpx  #30        ; utload ok for 1541 mode
     804D   FO 07          637            beq  1$
                           638
     804F   AD 180F        639            lda  pota1
     8052   29 20          640            and  #$20       ; 1/2 Mhz ?
     8054   FO OF          641            beq  realus     ; 1541 mode...ignore
                           642
     8056   A5 37          643    1$      lda  fastsr     ; clear clock & error return
     8058   29 EB          644            and  #$eb
     805A   85 37          645            sta  fastsr
```

```
Error Addr  Code           Seq    Source statement

                           646
     805C   BD 806E        647            lda  cmdctl,x    ; most sig bit set set error recover
     805F   8D 0202        648            sta  cmdbuf+2    ; save info here
     8062   6C 0075        649            jmp  (ip)
                           650
     8065   A9 EA          651    realus  lda  #<ublock    ; set default block add
     8067   85 6B          652            sta  usrjmp
     8069   A9 FF          653            lda  #>ublock
     806B   85 6C          654            sta  usrjmp+1
     806D   60             655    unused  rts
                           656
```

```
Error Addr  Code           Seq    Source statement

                           658
                           659    ; command tables and information
                           660
     806E   80 81 90       661    cmdctl    .byte  $80,$81,$90,$91,$b0,$b1,$f0,$f1,$00,$01,$B0,$01,$00,$01,$00,$01
     8071   91 B0 B1
     8074   F0 F1 00
     8077   01 B0 01
     807A   00 01 00
     807D   01
     807E   80 81 90       662              .byte  $80,$81,$90,$91,$b0,$b1,$f0,$f1,$00,$01,$B0,$01,$00,$01,$00,$80
     8081   91 B0 B1
     8084   F0 F1 00
     8087   01 B0 01
     808A   00 01 00
     808D   80
                           663
     808E   8371           664    cmdtbb    .word    fstrd          ; fast read drv #0 - 0000
     8090   837F           665              .word    ndkrd          ; fast read drv #1 - 0001
                           666
     8092   83EC           667              .word    fstwrt         ; fast write drv #0 - 0010
     8094   83F8           668              .word    ndkwrt         ; fast write drv #1 - 0011
                           669
     8096   848B           670              .word    fstsek         ; seek disk drv #0 - 0100
     8098   837F           671              .word    ndkrd          ; seek disk drv #1 - 0101
                           672
     809A   84B7           673              .word    fstfmt         ; format disk drv #0 - 0110
     809C   84B7           674              .word    fstfmt         ; format disk drv #1 - 0111
                           675
     809E   84F1           676              .word    cpmint         ; interleave disk drv #0 - 1000
     80A0   84F1           677              .word    cpmint         ; interleave disk drv #1 - 1001
                           678
     80A2   8517           679              .word    querdk         ; query disk format - 1010
     80A4   837F           680              .word    ndkrd          ; seek disk drv #1 - 1011
                           681
     80A6   856B           682              .word    inqst          ; return disk status - 1100
     80A8   837F           683              .word    ndkrd          ; return disk status - 1101
                           684
     80AA   85A5           685              .word    duplc1         ; backup drv0 to drv1 - 1110
     80AC   85A5           686              .word    duplc1         ; backup drv1 to drv0 - 1111
                           687
                           688    ; *****************************************************************
                           689
     80AE   8371           690              .word    fstrd          ; fast read drv #0 - 0000
     80B0   837F           691              .word    ndkrd          ; fast read drv #1 - 0001
                           692
     80B2   83EC           693              .word    fstwrt         ; fast write drv #0 - 0010
     80B4   83F8           694              .word    ndkwrt         ; fast write drv #1 - 0011
                           695
     80B6   848B           696              .word    fstsek         ; seek disk drv #0 - 0100
     80B8   837F           697              .word    ndkrd          ; seek disk drv #1 - 0101
```

```
Error Addr  Code           Seq    Source statement

                           698
     80BA   84B7           699             .word    fstfmt
     80BC   84B7           700             .word    fstfmt
                           701
     80BE   806D           702             .word    unused
     80C0   806D           703             .word    unused
                           704
     80C2   8517           705             .word    querdk          ; query disk format - 1010
     80C4   837F           706             .word    ndkrd           ; seek disk drv #1 - 1011
                           707
     80C6   806D           708             .word    unused
     80C8   806D           709             .word    unused
                           710
     80CA   8FE5           711             .word    chgutl
     80CC   9080           712             .word    fstload
                           713           .include   seratn1
```

```
Error Addr  Code           Seq    Source statement

                           715            .subttl 'seratn1.src'
                           716
     80CE   78             717    jatnsrv sei
     80CF   A9 00           718            lda  #0          ; clear pending flag
     80D1   85 7C           719            sta  atnpnd
     80D3   85 79           720            sta  lsnact      ; clear address flags
     80D5   85 7A           721            sta  tlkact
     80D7   A2 45           722            ldx  #topwrt     ; reset stack
     80D9   9A              723            txs
     80DA   20 81B2         724            jsr  spinp       ; serial port input
                           725
     80DD   A9 80           726            lda  #$80        ; set atn mode flag for acpt routine
     80DF   85 F8           727            sta  eoiflg      ; reset eoi flag to non-eoi state
     80E1   85 7D           728            sta  atnmod
                           729
     80E3   20 E9B7         730            jsr  clkhi
     80E6   20 E9A5         731            jsr  datlow      ; set data line low as response
                           732
     80E9   AD 1800         733            lda  pb          ; set atn ack to release data line
     80EC   09 10           734            ora  #atna
     80EE   8D 1800         735            sta  pb
                           736
     80F1   AD 1800         737    1$      lda  pb          ; test atn still here
     80F4   10 64           738            bpl  a2          ; gone !
                           739
     80F6   29 04           740            and  #clkin      ; clock still low
     80F8   D0 F7           741            bne  1$
                           742
     80FA   20 82CA         743    a3      jsr  jacptr+3    ; get a command byte
                           744
     80FD   C9 3F           745            cmp  #unlsn
     80FF   D0 0C           746            bne  a4
                           747
     8101   A5 37           748            lda  fastsr      ; clr fast host flag
     8103   29 BF           749            and  #$ff-$40
     8105   85 37           750            sta  fastsr
     8107   A9 00           751            lda  #0
     8109   85 79           752            sta  lsnact
     810B   F0 0E           753            beq  a6          ; jmp
                           754
     810D   C9 5F           755    a4      cmp  #untlk
     810F   D0 0D           756            bne  a5
                           757
     8111   A5 37           758            lda  fastsr      ; clr fast host flag
     8113   29 BF           759            and  #$ff-$40
     8115   85 37           760            sta  fastsr
     8117   A9 00           761            lda  #0
     8119   85 7A           762            sta  tlkact      ; untalk now
                           763
     811B   4C 8192         764    a6      jmp  a9          ; jmp
```

```
Error Addr  Code           Seq    Source statement

                           765
      811E  C5 78          766    a5      cmp  tlkadr      ; our talk address?
      8120  D0 0A          767            bne  a7          ; nope
                           768
      8122  A9 01          769            lda  #1          ; yes
      8124  85 7A          770            sta  tlkact      ; set active talker flag
      8126  A9 00          771            lda  #0
      8128  85 79          772            sta  lsnact      ; unlisten
      812A  F0 29          773            beq  a8          ; jump
                           774
      812C  C5 77          775    a7      cmp  lsnadr      ; our listen address?
      812E  D0 0A          776            bne  a10         ; nope
                           777
      8130  A9 01          778            lda  #1          ; yes
      8132  85 79          779            sta  lsnact
      8134  A9 00          780            lda  #0
      8136  85 7A          781            sta  tlkact
      8138  F0 1B          782            beq  a8          ; jump
                           783
      813A  AA             784    a10     tax              ; test if sa
      813B  29 60          785            and  #$60
      813D  C9 60          786            cmp  #$60        ; sa = $60 + n
      813F  D0 4C          787            bne  a11         ; did not get a valid command
                           788
      8141  8A             789            txa              ; a sa for me
      8142  85 84          790            sta  orgsa
      8144  29 0F          791            and  #$0f        ; strip junk
      8146  85 83          792            sta  sa
                           793
      8148  A5 84          794            lda  orgsa       ; test if close
      814A  29 F0          795            and  #$f0
      814C  C9 E0          796            cmp  #$e0
      814E  D0 42          797            bne  a9          ; no
                           798
      8150  58             799            cli
      8151  20 DAC0        800            jsr  close       ; close the file
      8154  78             801            sei
                           802
                           803    ;warning:::close doesn't return in time for a9
                           804
      8155  2C 1800        805    a8      bit  pb          ; test atn still here
      8158  30 A0          806            bmi  a3
                           807
                           808    ;atn gone , do what we where told to do
                           809
      815A  A9 00          810    a2      lda  #0          ; clear atn mode
      815C  85 7D          811            sta  atnmod
                           812
      815E  AD 1800        813            lda  pb          ; atn gone, release atn ack
      8161  29 EF          814            and  #$ff-atna
```

```
Error Addr  Code          Seq   Source statement

     8163   8D 1800       815           sta  pb
                          816
     8166   A5 79         817           lda  lsnact     ; listen ?
     8168   F0 0D         818           beq  a12
                          819
     816A   24 37         820           bit  fastsr     ; fast ?
     816C   50 03         821           bvc  a14
                          822
     816E   20 8199       823           jsr  reqfst     ; fast request
     8171   20 8342       824   a14     jsr  jlisten
     8174   4C 836B       825           jmp  xidle
                          826
     8177   A5 7A         827   a12     lda  tlkact     ; talk?
     8179   F0 0F         828           beq  a13
                          829
     817B   20 E99C       830           jsr  dathi      ; release data line
     817E   20 E9AE       831           jsr  clklow
                          832
     8181   20 A483       833           jsr  jslowd     ; slow down for plus4 series
     8184   20 81EA       834           jsr  jtalk
     8187   20 A483       835           jsr  jslowd     ; slow down for plus4 series
     818A   4C 8366       836   a13     jmp  jilerr     ; release all lines and go to idle
                          837
                          838   ;fix so (device not present) errors reported
                          839
     818D   A9 10         840   a11     lda  #$10       ; kill all
     818F   8D 1800       841           sta  pb
                          842
     8192   2C 1800       843   a9      bit  pb
     8195   10 C3         844           bpl  a2         ; exit out same way after atn done
                          845
     8197   30 F9         846           bmi  a9         ; bra
                          847
     8199   20 EA59       848   reqfst  jsr  tstatn     ; does the host want us ?
     819C   20 E9C0       849           jsr  debnc
     819F   29 04         850           and  #clkin
     81A1   D0 F6         851           bne  reqfst     ; wait for clk hi
                          852
     81A3   20 81CE       853           jsr  spout      ; output
                          854
     81A6   A9 00         855           lda  #0
     81A8   8D 400C       856           sta  sdr        ; send zero
                          857
     81AB   A9 08         858   slowb   lda  #8
     81AD   2C 400D       859   1$      bit  icr
     81B0   F0 FB         860           beq  1$
                          861
     81B2   08            862   spinp   php             ; save uP status
     81B3   78            863           sei
     81B4   AD 400E       864           lda  cra        ; turn 6526 in
```

```
Error Addr  Code           Seq    Source statement

     81B7   29 BF          865            and  #%10111111 ; serial port input
     81B9   8D 400E        866            sta  cra
     81BC   AD 180F        867            lda  pota1      ; turn drvr in
     81BF   29 FD          868            and  #%11111101
     81C1   8D 180F        869            sta  pota1
     81C4   A9 84          870            lda  #$84
     81C6   8D 400D        871            sta  icr        ; enable irq from serial port
     81C9   2C 400D        872            bit  icr        ; clear pending
     81CC   28             873            plp
     81CD   60             874            rts
                           875
     81CE   08             876    spout   php             ; save uP status
     81CF   78             877            sei
     81D0   AD 180F        878            lda  pota1
     81D3   09 02          879            ora  #%00000010 ; turn drvr out
     81D5   8D 180F        880            sta  pota1
     81D8   AD 400E        881            lda  cra        ; turn 6526 out
     81DB   09 40          882            ora  #%01000000
     81DD   8D 400E        883            sta  cra
     81E0   A9 08          884            lda  #8
     81E2   8D 400D        885            sta  icr        ; disable irq from serial port
     81E5   2C 400D        886            bit  icr        ; clr pending
     81E8   28             887            plp
     81E9   60             888            rts
                           889            .include  sertalk1
```

```
Error Addr  Code           Seq    Source statement

                           891            .subttl 'sertalk1.src'
                           892
     81EA   78             893    jtalk   sei             ; find if open channel
     81EB   20 D0EB         894            jsr  fndrch
     81EE   B0 06           895            bcs  1$         ; no one home
                           896
     81F0   A6 82           897    2$      ldx  lindx
     81F2   B5 F2           898            lda  chnrdy,x
     81F4   30 01           899            bmi  3$
                           900
     81F6   60              901    1$      rts
                           902
                           903    ;  code added to correct verify error
                           904
     81F7   20 EA59         905    3$      jsr  tstatn     ; test for atn
     81FA   20 E9C0         906            jsr  debnc      ; debounce
     81FD   29 01           907            and  #datin
     81FF   08              908            php
     8200   20 E9B7         909            jsr  clkhi      ; set clk hi
     8203   28              910            plp             ; see if verify error...
     8204   F0 12           911            beq  4$         ; br,  yes...data line hi, eoi !!!!
                           912
     8206   20 EA59         913    5$      jsr  tstatn     ; test for atn
     8209   20 E9C0         914            jsr  debnc
     820C   29 01           915            and  #datin
     820E   D0 F6           916            bne  5$         ; wait for data high
                           917
     8210   A6 82           918            ldx  lindx      ; prepare to send eoi if needed
     8212   B5 F2           919            lda  chnrdy,x
     8214   29 08           920            and  #eoi
     8216   D0 14           921            bne  7$         ; no eoi
                           922
     8218   20 EA59         923    4$      jsr  tstatn     ; test for atn
     821B   20 E9C0         924            jsr  debnc      ; debounce
     821E   29 01           925            and  #datin     ; test if data line is low
     8220   D0 F6           926            bne  4$         ; yes, wait till hi
                           927
     8222   20 EA59         928    6$      jsr  tstatn     ; test for atn
     8225   20 E9C0         929            jsr  debnc      ; debounce
     8228   29 01           930            and  #datin
     822A   F0 F6           931            beq  6$
                           932
     822C   20 E9AE         933    7$      jsr  clklow     ; set clock low
     822F   20 EA59         934            jsr  tstatn     ; chk atn line
     8232   20 E9C0         935            jsr  debnc      ; debounce
     8235   29 01           936            and  #datin
     8237   D0 F3           937            bne  7$
                           938
                           939    ; ***************  fast routines  *****************
                           940
```

```
Error Addr  Code          Seq   Source statement

     8239   24 37         941           bit  fastsr      ; fast or slow ?
     823B   50 39         942           bvc  10$
                          943
     823D   AD 180F       944           lda  pota1
     8240   09 02         945           ora  #%00000010 ; turn drvr out
     8242   8D 180F       946           sta  pota1
                          947
     8245   AD 400E       948           lda  cra         ; turn 6526 out
     8248   09 40         949           ora  #%01000000 ; go output mode
     824A   8D 400E       950           sta  cra
     824D   2C 400D       951           bit  icr         ; clear pending
     8250   A6 82         952           ldx  lindx
     8252   BD 023E       953           lda  chndat,x    ; get data
     8255   8D 400C       954           sta  sdr         ; send it
     8258   AD 400D       955   8$      lda  icr         ; wait for byte ready
     825B   29 08         956           and  #8          ; chk sr bit
     825D   F0 F9         957           beq  8$
     825F   AD 400E       958           lda  cra
     8262   29 BF         959           and  #%10111111 ; release the data bus
     8264   8D 400E       960           sta  cra         ; must go input
     8267   AD 180F       961           lda  pota1
     826A   29 FD         962           and  #%11111101 ; turn drvr in
     826C   8D 180F       963           sta  pota1
     826F   A9 84         964           lda  #$84
     8271   8D 400D       965           sta  icr         ; enable irq
     8274   D0 3C         966           bne  9$          ; wait for data accepted
                          967
                          968   ; ****************    end fast     *****************
                          969
     8276                 970   10$
     8276   A9 08         971           lda  #8          ; set up bit counter
     8278   85 98         972           sta  cont
                          973
     827A                 974   11$
     827A   20 E9C0       975           jsr  debnc       ; let port settle
                          976
     827D   29 01         977           and  #datin      ; test that data line is now high before we send
     827F   D0 43         978           bne  16$
                          979
     8281   A6 82         980   12$     ldx  lindx       ; get byte to send
     8283   BD 023E       981           lda  chndat,x
     8286   6A            982           ror  a
     8287   9D 023E       983           sta  chndat,x
                          984
     828A   B0 05         985           bcs  13$         ; send a 1
                          986
     828C   20 E9A5       987           jsr  datlow      ; send a 0
     828F   D0 03         988           bne  14$         ; and clock it
                          989
     8291   20 E99C       990   13$     jsr  dathi
```

```
Error Addr  Code           Seq    Source statement

     8294   20 A47E        991   14$      jsr  jslower     ; wait a bit for t-s ( 2 mhz )
     8297   A5 23          992             lda  drvtrk+1
     8299   DO E6          993             bne  12$
                           994
     829B   20 A483        995             jsr  jslowd      ; delay 40 us ( host dma )
     829E   20 E9B7        996             jsr  clkhi       ; rising edge clock
     82A1   20 A47E        997             jsr  jslower     ; increase t-v ( 2 mhz )
     82A4   A5 23          998             lda  drvtrk+1    ; check speed flag
     82A6   DO O3          999             bne  15$
                          1000
     82A8   20 A483       1001             jsr  jslowd      ; delay 40 us ( host dma )
     82AB   20 FEFB       1002   15$       jsr  clkdat      ; pull clock low and release data
                          1003
     82AE   C6 98         1004             dec  cont        ; more bits?
     82BO   DO C8         1005             bne  11$         ; yes
                          1006
     82B2   20 EA59       1007   9$        jsr  tstatn      ; test for atn
     82B5   20 E9CO       1008             jsr  debnc       ; debounce
     82B8   29 01         1009             and  #datin
     82BA   FO F6         1010             beq  9$          ; wait for data low
                          1011
     82BC   58            1012             cli              ; let the controller run
     82BD   20 D3AA       1013             jsr  get         ; get the next byte
     82CO   78            1014             sei              ; sorry sync protocol
                          1015
     82C1   4C 81FO       1016             jmp  2$          ; keep on talkin
                          1017
     82C4   4C 8362       1018   16$       jmp  jfrmerr
                          1019             .include  serlistn1
```

```
Error Addr  Code           Seq    Source statement

                          1021            .subttl 'serlistn1.src'
                          1022
     82C7   2C 400D       1023    jacptr  bit  icr         ; clear pending
     82CA   A9 08         1024            lda  #8          ; set byte bit count
     82CC   85 98         1025            sta  cont
                          1026
     82CE   20 EA59       1027    1$      jsr  tstatn
     82D1   20 E9C0       1028            jsr  debnc
     82D4   29 04         1029            and  #clkin
     82D6   D0 F6         1030            bne  1$
                          1031
     82D8   20 E99C       1032            jsr  dathi       ; make data line hi
                          1033
     82DB   A9 01         1034            lda  #1
     82DD   2C 1800       1035    2$      bit  pb          ; wait for data high
     82E0   D0 FB         1036            bne  2$
                          1037
     82E2   8D 1805       1038            sta  t1hc1       ; wait for 256 uS
                          1039
     82E5   20 EA59       1040    3$      jsr  tstatn
     82E8   AD 180D       1041            lda  ifr1
     82EB   29 40         1042            and  #$40        ; time out ?
     82ED   D0 09         1043            bne  4$
                          1044
     82EF   20 E9C0       1045            jsr  debnc       ; test clock low
     82F2   29 04         1046            and  #clkin
     82F4   F0 EF         1047            beq  3$          ; no
     82F6   D0 19         1048            bne  5$          ; yes
                          1049
     82F8   20 E9A5       1050    4$      jsr  datlow      ; set data line low as response
                          1051
     82FB   A2 18         1052            ldx  #24         ; delay for talker turnaround
     82FD   CA            1053    6$      dex
     82FE   D0 FD         1054            bne  6$
                          1055
     8300   20 E99C       1056            jsr  dathi       ; set data line hi
                          1057
     8303   20 EA59       1058    7$      jsr  tstatn
     8306   20 E9C0       1059            jsr  debnc       ; wait for low clock
     8309   29 04         1060            and  #clkin
     830B   F0 F6         1061            beq  7$
                          1062
     830D   A9 00         1063            lda  #0          ; set eoi received
     830F   85 F8         1064            sta  eoiflg
                          1065
     8311                 1066    5$
     8311   AD 1800       1067    8$      lda  pb          ; wait for clock high
     8314   49 01         1068            eor  #01         ; complement datain
                          1069    ;**********  fast routines  **********
     8316   AA            1070            tax              ; save .a
```

```
Error Addr  Code           Seq    Source statement

     8317   AD 400D       1071           lda  icr
     831A   29 08         1072           and  #8          ; fast byte ?
     831C   F0 08         1073           beq  11$
                          1074
     831E   AD 400C       1075           lda  sdr         ; get data
     8321   85 85         1076           sta  data        ; keep
     8323   4C 833C       1077           jmp  10$         ; finish up
                          1078  ;**********    end fast     **********
     8326   8A            1079  11$      txa              ; restore .a
     8327   4A            1080           lsr  a           ; shift into carry
     8328   29 02         1081           and  #$02        ; clkin/2
     832A   D0 E5         1082           bne  8$
                          1083
     832C   66 85         1084           ror  data
                          1085
     832E   20 EA59       1086  9$       jsr  tstatn
     8331   20 E9C0       1087           jsr  debnc
     8334   29 04         1088           and  #clkin      ; wait for clock low
     8336   F0 F6         1089           beq  9$
                          1090
     8338   C6 98         1091           dec  cont        ; more to do?
     833A   D0 D5         1092           bne  8$
                          1093
     833C   20 E9A5       1094  10$      jsr  datlow      ; set data line low
     833F   A5 85         1095           lda  data
     8341   60            1096           rts
                          1097
     8342   78            1098  jlisten sei
                          1099
     8343   20 D107       1100           jsr  fndwch      ; test if active write channel
     8346   B0 05         1101           bcs  1$
                          1102
     8348   B5 F2         1103           lda  chnrdy,x
     834A   6A            1104           ror  a
     834B   B0 0B         1105           bcs  2$
                          1106
     834D   A5 84         1107  1$       lda  orgsa       ; test if open
     834F   29 F0         1108           and  #$f0
     8351   C9 F0         1109           cmp  #$f0
     8353   F0 03         1110           beq  2$          ; its an open
                          1111
     8355   4C 8366       1112           jmp  jilerr      ; not active channel
                          1113
     8358   20 82C7       1114  2$       jsr  jacptr      ; get a byte
                          1115
     835B   58            1116           cli
     835C   20 CFB7       1117           jsr  put         ; put(data,eoiflg,sa)
                          1118
     835F   4C 8342       1119           jmp  jlisten     ; and keep on listen
                          1120
```

```
Error Addr  Code           Seq    Source statement

                          1121
     8362                 1122   jfrmerr
     8362   A9 00         1123           lda  #0          ; *** rom ds -06 *** / clr fastsr after serial bus error
     8364   85 37         1124           sta  fastsr
     8366                 1125   jilerr
     8366   A9 00         1126           lda  #0          ; release all bus lines
     8368   8D 1800       1127           sta  pb          ; in atnmod
     836B   20 81B2       1128   xidle   jsr  spinp
     836E   4C EBE7       1129           jmp  idle        ; go idle it
                          1130           .include  fastutl
```

```
Error Addr  Code          Seq   Source statement

                         1132           .subttl 'fastutl.src'
                         1133   ;NOTE: ALL BURST COMMANDS ARE SENT VIA KERNAL I/O CALLS.
                         1134   ;
                         1135   ;BURST CMD ONE - READ
                         1136   ;
                         1137   ;     BYTE  BIT 7       6       5       4       3       2       1       0
                         1138   ;=============================================================================
                         1139   ;       00      0       1       0       1       0       1       0       1
                         1140   ;-----------------------------------------------------------------------------
                         1141   ;       01      0       0       1       1       0       0       0       0
                         1142   ;-----------------------------------------------------------------------------
                         1143   ;       02      T       E       B       S       0       0       0       N
                         1144   ;-----------------------------------------------------------------------------
                         1145   ;       03                      DESTINATION TRACK
                         1146   ;-----------------------------------------------------------------------------
                         1147   ;       04                      DESTINATION SECTOR
                         1148   ;-----------------------------------------------------------------------------
                         1149   ;       05                      NUMBER OF SECTORS
                         1150   ;-----------------------------------------------------------------------------
                         1151   ;       06                      NEXT TRACK (OPTIONAL)
                         1152   ;-----------------------------------------------------------------------------
                         1153   ;
                         1154   ;
                         1155   ;   RANGE:
                         1156   ;
                         1157   ;       MFM ALL VALUES ARE DETERMINED UPON THE PARTICULAR DISK FORMAT.
                         1158   ;       GCR SEE SOFTWARE SPECIFICATIONS.
                         1159   ;
                         1160   ;
                         1161   ;SWITCHES:
                         1162   ;
                         1163   ;       T - TRANSFER DATA (1=NO TRANSFER)
                         1164   ;       E - IGNORE ERROR (1=IGNORE)
                         1165   ;       B - BUFFER TRANSFER ONLY (1=BUFFER TRANSFER ONLY)
                         1166   ;       S - SIDE SELECT (MFM ONLY)
                         1167   ;       N - DRIVE NUMBER
                         1168   ;
                         1169   ;
                         1170   ;
                         1171   ;PROTOCOL:
                         1172   ;
                         1173   ;       BURST HANDSHAKE.
                         1174   ;
                         1175   ;
                         1176   ;
                         1177   ;CONVENTIONS:
                         1178   ;
                         1179   ;       CMD ONE MUST BE PRECEEDED  WITH CMD 3 OR CMD 6 ONCE  AFTER
                         1180   ;       A NEW DISK IS INSERTED.
                         1181   ;
```

```
Error Addr  Code           Seq    Source statement

                          1182    ;
                          1183    ;OUTPUT:
                          1184    ;       ONE  BURST  STATUS  BYTE PRECEEDING BURST DATA WILL BE SENT
                          1185    ;       FOR  EVERY  SECTOR  TRANSFERED.    ON  AN  ERROR  CONDITION
                          1186    ;       DATA  WILL NOT BE SENT UNLESS THE E BIT IS SET.
```

```
Error Addr  Code          Seq   Source statement

                          1188  ;BURST CMD TWO - WRITE
                          1189  ;
                          1190  ;     BYTE  BIT 7       6       5       4       3       2       1       0
                          1191  ;=============================================================================
                          1192  ;       00      0       1       0       1       0       1       0       1
                          1193  ;-----------------------------------------------------------------------------
                          1194  ;       01      0       0       1       1       0       0       0       0
                          1195  ;-----------------------------------------------------------------------------
                          1196  ;       02      T       E       B       S       0       0       1       N
                          1197  ;-----------------------------------------------------------------------------
                          1198  ;       03                      DESTINATION TRACK
                          1199  ;-----------------------------------------------------------------------------
                          1200  ;       04                      DESTINATION SECTOR
                          1201  ;-----------------------------------------------------------------------------
                          1202  ;       05                      NUMBER OF SECTORS
                          1203  ;-----------------------------------------------------------------------------
                          1204  ;       06                      NEXT TRACK (OPTIONAL)
                          1205  ;-----------------------------------------------------------------------------
                          1206  ;
                          1207  ;
                          1208  ;   RANGE:
                          1209  ;
                          1210  ;       MFM ALL VALUES ARE DETERMINED UPON THE PARTICULAR DISK FORMAT.
                          1211  ;       GCR SEE SOFTWARE SPECIFICATIONS.
                          1212  ;
                          1213  ;
                          1214  ;SWITCHES:
                          1215  ;
                          1216  ;       T - TRANSFER DATA (1=NO TRANSFER)
                          1217  ;       E - IGNORE ERROR (1=IGNORE)
                          1218  ;       B - BUFFER TRANSFER ONLY (1=BUFFER TRANSFER ONLY)
                          1219  ;       S - SIDE SELECT (MFM ONLY)
                          1220  ;       N - DRIVE NUMBER
                          1221  ;
                          1222  ;
                          1223  ;
                          1224  ;PROTOCOL:
                          1225  ;
                          1226  ;       BURST HANDSHAKE.
                          1227  ;
                          1228  ;
                          1229  ;
                          1230  ;CONVENTIONS:
                          1231  ;
                          1232  ;       CMD TWO MUST BE PRECEEDED  WITH CMD 3 OR CMD 6 ONCE  AFTER
                          1233  ;       A NEW DISK IS INSERTED.
                          1234  ;
                          1235  ;
                          1236  ;INPUT:
                          1237  ;       HOST MUST TRANSFER BURST DATA.
```

```
Error Addr  Code           Seq    Source statement

                          1238    ;
                          1239    ;
                          1240    ;OUTPUT:
                          1241    ;       ONE BURST STATUS  BYTE FOLLOWING EACH WRITE OPERATION.
```

```
Error Addr  Code          Seq   Source statement

                          1243  ;BURST CMD THREE - INQUIRE DISK
                          1244  ;
                          1245  ;
                          1246  ;     BYTE  BIT 7       6       5       4       3       2       1       0
                          1247  ;=============================================================================
                          1248  ;       00      0       1       0       1       0       1       0       1
                          1249  ;-----------------------------------------------------------------------------
                          1250  ;       01      0       0       1       1       0       0       0       0
                          1251  ;-----------------------------------------------------------------------------
                          1252  ;       02      X       X       X       S       0       1       0       N
                          1253  ;-----------------------------------------------------------------------------
                          1254  ;
                          1255  ;
                          1256  ;
                          1257  ;SWITCHES:,
                          1258  ;
                          1259  ;       S - SIDE SELECT (MFM ONLY)
                          1260  ;       N - DRIVE NUMBER
                          1261  ;
                          1262  ;
                          1263  ;OUTPUT:
                          1264  ;       ONE BURST STATUS  BYTE FOLLOWING THE INQUIRE OPERATION.
```

```
Error Addr  Code          Seq   Source statement

                          1266  ;BURST CMD FOUR - FORMAT MFM
                          1267  ;
                          1268  ;
                          1269  ;     BYTE  BIT 7      6      5      4      3      2      1      0
                          1270  ;=============================================================================
                          1271  ;       00      0      1      0      1      0      1      0      1
                          1272  ;-----------------------------------------------------------------------------
                          1273  ;       01      0      0      1      1      0      0      0      0
                          1274  ;-----------------------------------------------------------------------------
                          1275  ;       02      P      I      D      S      0      1      1      N
                          1276  ;-----------------------------------------------------------------------------
                          1277  ;       03     M=1     T                LOGICAL STARTING SECTOR
                          1278  ;-----------------------------------------------------------------------------
                          1279  ;       04            INTERLEAVE           (OPTIONAL DEF-0)
                          1280  ;-----------------------------------------------------------------------------
                          1281  ;       05            SECTOR SIZE        * (OPTIONAL 01 for 256 Byte Sectors)
                          1282  ;-----------------------------------------------------------------------------
                          1283  ;       06            LAST TRACK NUMBER    (OPTIONAL DEF-39)
                          1284  ;-----------------------------------------------------------------------------
                          1285  ;       07            NUMBER OF SECTORS    (OPTIONAL DEPENDS ON BYTE 05)
                          1286  ;-----------------------------------------------------------------------------
                          1287  ;       08          LOGICAL STARTING TRACK (OPTIONAL DEF-0)
                          1288  ;-----------------------------------------------------------------------------
                          1289  ;       09            STARTING TRACK OFFSET (OPTIONAL DEF-0)
                          1290  ;-----------------------------------------------------------------------------
                          1291  ;       0A            FILL BYTE            (OPTIONAL DEF-$E5)
                          1292  ;-----------------------------------------------------------------------------
                          1293  ;       0B-??         SECTOR TABLE         (OPTIONAL T-BIT SET)
                          1294  ; -----------------------------------------------------------------------------
                          1295  ;
                          1296  ;SWITCHES:
                          1297  ;
                          1298  ;       P - PARTIAL FORMAT (1=PARTIAL)
                          1299  ;       I - INDEX ADDRESS MARK WRITTEN (1=WRITTEN)
                          1300  ;       D - DOUBLE SIDED FLAG (1=FORMAT DOUBLE SIDED)
                          1301  ;       S - SIDE SELECT
                          1302  ;       T - SECTOR TABLE INCLUDED (1=INCLUDED, ALL OTHER PARMS
                          1303  ;           MUST BE INCLUDED)
                          1304  ;       N - DRIVE NUMBER
                          1305  ;
                          1306  ;
                          1307  ;PROTOCOL:
                          1308  ;
                          1309  ;       CONVENTIONAL.
                          1310  ;
                          1311  ;
                          1312  ;
                          1313  ;CONVENTIONS:
                          1314  ;
                          1315  ;       AFTER FORMATTING CMD 3 MUST BE SENT ON CONVENTIONAL FORMAT
```

```
Error Addr  Code          Seq   Source statement

                         1316   ;       OR CMD 6 ON A NON-CONVENTIONAL FORMAT.
                         1317   ;
                         1318   ;
                         1319   ;OUTPUT:
                         1320   ;         NONE.  STATUS WILL BE UPDATED WITHIN THE DRIVE.
                         1321   ;
                         1322   ; * 00 - 128 byte sectors
                         1323   ;   01 - 256 byte sectors
                         1324   ;   02 - 512 byte sectors
                         1325   ;   03 - 1024 byte sectors
```

```
Error Addr  Code          Seq    Source statement

                         1327   ;BURST CMD FOUR - FORMAT GCR (NO DIRECTORY)
                         1328   ;
                         1329   ;
                         1330   ;     BYTE  BIT 7       6       5       4       3       2       1       0
                         1331   ;=============================================================================
                         1332   ;       00      0       1       0       1       0       1       0       1
                         1333   ;-----------------------------------------------------------------------------
                         1334   ;       01      0       0       1       1       0       0       0       0
                         1335   ;-----------------------------------------------------------------------------
                         1336   ;       02      X       X       X       X       0       1       1       N
                         1337   ;-----------------------------------------------------------------------------
                         1338   ;       03      M=0
                         1339   ;-----------------------------------------------------------------------------
                         1340   ;       04                      ID LOW
                         1341   ;-----------------------------------------------------------------------------
                         1342   ;       05                      ID HIGH
                         1343   ;-----------------------------------------------------------------------------
                         1344   ;
                         1345   ;
                         1346   ;SWITCHES:
                         1347   ;
                         1348   ;       N - DRIVE NUMBER
                         1349   ;       X - DON'T CARE.
                         1350   ;
                         1351   ;PROTOCOL:
                         1352   ;
                         1353   ;       CONVENTIONAL.
                         1354   ;
                         1355   ;
                         1356   ;CONVENTIONS:
                         1357   ;
                         1358   ;       AFTER FORMATTING CMD 3 MUST BE SENT.
                         1359   ;
                         1360   ;       NOTE: THIS  COMMAND  DOES NOT WRITE THE BAM OR THE
                         1361   ;       DIRECTORY ENTRIES  ( DOUBLE SIDED FLAG WILL NOT BE
                         1362   ;       ON TRACK 18, SECTOR 0).  IT  IS SUGGESTED THAT THE
                         1363   ;       CONVENTIONAL 'NEW' FORMAT COMMAND BE USED.
                         1364   ;
                         1365   ;OUTPUT:
                         1366   ;        NONE.  STATUS WILL BE UPDATED WITHIN THE DRIVE.
```

```
Error Addr  Code          Seq   Source statement

                         1368   ;BURST CMD FIVE - SECTOR INTERLEAVE
                         1369   ;
                         1370   ;
                         1371   ;     BYTE  BIT 7       6       5       4       3       2       1       0
                         1372   ;=============================================================================
                         1373   ;       00      0       1       0       1       0       1       0       1
                         1374   ;-----------------------------------------------------------------------------
                         1375   ;       01      0       0       1       1       0       0       0       0
                         1376   ;-----------------------------------------------------------------------------
                         1377   ;       02      W       X       X       0       1       0       0       N
                         1378   ;-----------------------------------------------------------------------------
                         1379   ;       04                      INTERLEAVE
                         1380   ;-----------------------------------------------------------------------------
                         1381   ;
                         1382   ;
                         1383   ;SWITCHES:
                         1384   ;
                         1385   ;       W - WRITE SWITCH
                         1386   ;       N - DRIVE NUMBER
                         1387   ;       X - DON'T CARE
                         1388   ;
                         1389   ;PROTOCOL:
                         1390   ;
                         1391   ;       CONVENTIONAL, UNLESS THE W-BIT IS SET.
                         1392   ;
                         1393   ;
                         1394   ;CONVENTIONS:
                         1395   ;
                         1396   ;       THIS IS A SOFT INTERLEAVE USED FOR MULTI-SECTOR BURST
                         1397   ;       READ AND WRITE.
                         1398   ;
                         1399   ;OUTPUT:
                         1400   ;        NONE (W=0), INTERLEAVE BURST BYTE (W=1)
```

```
Error Addr  Code          Seq   Source statement

                          1402  ;BURST CMD SIX - QUERY DISK FORMAT
                          1403  ;
                          1404  ;
                          1405  ;     BYTE  BIT 7       6       5       4       3       2       1       0
                          1406  ;=============================================================================
                          1407  ;       00      0       1       0       1       0       1       0       1
                          1408  ;-----------------------------------------------------------------------------
                          1409  ;       01      0       0       1       1       0       0       0       0
                          1410  ;-----------------------------------------------------------------------------
                          1411  ;       02      F       X       X       S       1       0       1       N
                          1412  ;-----------------------------------------------------------------------------
                          1413  ;       03                  OFFSET      (OPTIONAL F-BIT SET)
                          1414  ;-----------------------------------------------------------------------------
                          1415  ;
                          1416  ;
                          1417  ;SWITCHES:
                          1418  ;
                          1419  ;       F - FORCE FLAG (F=1, WILL STEP THE HEAD WITH THE
                          1420  ;           OFFSET SPECIFIED IN BYTE 03.
                          1421  ;       N - DRIVE NUMBER
                          1422  ;       X - DON'T CARE.
                          1423  ;
                          1424  ;PROTOCOL:
                          1425  ;
                          1426  ;       CONVENTIONAL.
                          1427  ;
                          1428  ;
                          1429  ;CONVENTIONS:
                          1430  ;
                          1431  ;       THIS IS A METHOD OF DETERMINING THE FORMAT OF THE DISK ON
                          1432  ;       ANY PARTICULAR TRACK.
                          1433  ;
                          1434  ;
                          1435  ;OUTPUT:
                          1436  ;       BURST STATUS BYTE (IF THERE WAS AN ERROR OR IF THE
                          1437  ;                          FORMAT IS GCR NO BYTES WILL FOLLOW)
                          1438  ;       BURST STATUS BYTE (IF THERE WAS AN ERROR IN COMPILING
                          1439  ;                          MFM FORMAT INFORMATION NO BYTES
                          1440  ;                          WILL FOLLOW)
                          1441  ;       NUMBER OF SECTORS (THE NUMBER OF SECTORS ON A PARTICULAR
                          1442  ;                          TRACK)
                          1443  ;       LOGICAL TRACK     (THE LOGICAL TRACK NUMBER FOUND IN THE
                          1444  ;                          DISK HEADER)
                          1445  ;       MINIMUM SECTOR    (THE LOGICAL SECTOR WITH THE LOWEST
                          1446  ;                          VALUE ADDRESS)
                          1447  ;       MAXIMUM SECTOR    (THE LOGICAL SECTOR WITH THE HIGHEST
                          1448  ;                          VALUE ADDRESS)
                          1449  ;       CP/M INTERLEAVE   (THE HARD INTERLEAVE FOUND ON A PARTICULAR
                          1450  ;                          TRACK)
                          1451  ;
```

```
Error Addr  Code          Seq   Source statement

                         1453   ;BURST CMD SEVEN - INQUIRE STATUS
                         1454   ;
                         1455   ;
                         1456   ;     BYTE  BIT 7       6       5       4       3       2       1       0
                         1457   ;=============================================================================
                         1458   ;       00      0       1       0       1       0       1       0       1
                         1459   ;-----------------------------------------------------------------------------
                         1460   ;       01      0       0       1       1       0       0       0       0
                         1461   ;-----------------------------------------------------------------------------
                         1462   ;       02      W       C       X       0       1       1       0       N
                         1463   ;-----------------------------------------------------------------------------
                         1464   ;       03                      NEW STATUS (W-BIT CLEAR)
                         1465   ;-----------------------------------------------------------------------------
                         1466   ;
                         1467   ;SWITCHES:
                         1468   ;
                         1469   ;       W - WRITE SWITCH
                         1470   ;       C - CHANGE (C=1 & W=0 - FORCE LOGIN DISK,
                         1471   ;                   C=1 & W=1 - RETURN WHETHER DISK WAS LOGGED
                         1472   ;                   IE. $XB ERROR OR OLD STATUS)
                         1473   ;       N - DRIVE NUMBER
                         1474   ;       X - DON'T CARE
                         1475   ;
                         1476   ;PROTOCOL:
                         1477   ;
                         1478   ;       BURST.
                         1479   ;
                         1480   ;
                         1481   ;CONVENTIONS:
                         1482   ;
                         1483   ;       THIS IS A METHOD OF READING OR WRITING CURRENT STATUS.
                         1484   ;
                         1485   ;
                         1486   ;OUTPUT:
                         1487   ;        NONE (W=0), BURST STATUS BYTE (W=1)
                         1488   ;
                         1489   ;
```

```
Error Addr  Code          Seq   Source statement

                          1491  ;BURST CMD EIGHT - BACKUP DISK
                          1492  ;
                          1493  ;     BYTE  BIT 7       6       5       4       3       2       1       0
                          1494  ;=============================================================================
                          1495  ;       00      0       1       0       1       0       1       0       1
                          1496  ;-----------------------------------------------------------------------------
                          1497  ;       01      0       0       1       1       0       0       0       0
                          1498  ;-----------------------------------------------------------------------------
                          1499  ;       02      ?       ?       ?       ?       1       1       1       ?
                          1500  ;-----------------------------------------------------------------------------
                          1501  ;
                          1502  ;SWITCHES:
                          1503  ;
                          1504  ;       ? - GOT ME
                          1505  ;
```

```
Error Addr  Code          Seq   Source statement

                         1507   ;BURST CMD NINE - CHGUTL UTILITY
                         1508   ;
                         1509   ;
                         1510   ;     BYTE  BIT 7       6       5       4       3       2       1       0
                         1511   ;=============================================================================
                         1512   ;       00      0       1       0       1       0       1       0       1
                         1513   ;-----------------------------------------------------------------------------
                         1514   ;       01      0       0       1       1       0       0       0       0
                         1515   ;-----------------------------------------------------------------------------
                         1516   ;       02      X       X       X       1       1       1       1       0
                         1517   ;-----------------------------------------------------------------------------
                         1518   ;       03          UTILITY COMMANDS: 'S', 'R', 'T', 'M', 'H', #DEV
                         1519   ;-----------------------------------------------------------------------------
                         1520   ;       04          COMMAND PARAMETER
                         1521   ;-----------------------------------------------------------------------------
                         1522   ;SWITCHES:
                         1523   ;
                         1524   ;       X - DON'T CARE.
                         1525   ;
                         1526   ;
                         1527   ;
                         1528   ;UTILITY COMMANDS:
                         1529   ;
                         1530   ;       'S' - DOS SECTOR INTERLEAVE.
                         1531   ;       'R' - DOS RETRIES.
                         1532   ;       'T' - ROM SIGNATURE ANALYSIS.
                         1533   ;       'M' - MODE SELECT.
                         1534   ;       'H' - HEAD SELECT.
                         1535   ;      #DEV - DEVICE #.
                         1536   ;
                         1537   ;
                         1538   ;NOTE: BYTE 02 IS EQUIVALENT TO A '>'
                         1539   ;
                         1540   ;EXAMPLES:
                         1541   ;       "U0>S"+CHR$(SECTOR-INTERLEAVE)
                         1542   ;       "U0>R"+CHR$(RETRIES)
                         1543   ;       "U0>T" (On failure LED will blink 4 times)
                         1544   ;       "U0>M1"=1571 MODE, "U0>M0"=1541 MODE
                         1545   ;       "U0>H0"=SIDE ZERO, "U0>H1"=SIDE ONE (1541 MODE ONLY)
                         1546   ;       "U0>"+CHR$(#DEV), WHERE #DEV = 8-30
```

```
Error Addr  Code          Seq   Source statement

                         1548   ;BURST CMD TEN - FASTLOAD UTILITY
                         1549   ;
                         1550   ;
                         1551   ;     BYTE  BIT 7       6       5       4       3       2       1       0
                         1552   ;=============================================================================
                         1553   ;       00      0       1       0       1       0       1       0       1
                         1554   ;-----------------------------------------------------------------------------
                         1555   ;       01      0       0       1       1       0       0       0       0
                         1556   ;-----------------------------------------------------------------------------
                         1557   ;       02      P       X       X       1       1       1       1       1
                         1558   ;-----------------------------------------------------------------------------
                         1559   ;       03                      FILE NAME
                         1560   ;-----------------------------------------------------------------------------
                         1561   ;
                         1562   ;SWITCHES:
                         1563   ;
                         1564   ;       P - SEQUENTIAL FILE BIT (P=1, DOES NOT HAVE TO BE A PROGRAM
                         1565   ;                                      FILE)
                         1566   ;       X - DON'T CARE.
                         1567   ;
                         1568   ;PROTOCOL:
                         1569   ;
                         1570   ;       BURST.
                         1571   ;
                         1572   ;
                         1573   ;OUTPUT:
                         1574   ;        BURST STATUS BYTE PRECEEDING EACH SECTOR TRANSFERED.
                         1575   ;
                         1576   ;STATUS IS AS FOLLOWS:
                         1577   ;
                         1578   ;       0000000X ..... OK
                         1579   ;     * 00000010 ..... FILE NOT FOUND
                         1580   ;       00011111 ..... EOI
                         1581   ;
                         1582   ;* VALUES BETWEEN THE RANGE 3-15 SHOULD BE CONSIDERED A FILE READ ERROR.
```

```
Error Addr  Code          Seq   Source statement

                          1584  ;STATUS BYTE BREAK DOWN
                          1585  ;
                          1586  ;
                          1587  ;      BIT 7       6       5       4       3       2       1       0
                          1588  ;=============================================================================
                          1589  ;        MODE      DN     SECTOR SIZE      [  CONTROLLER  STATUS   ]
                          1590  ;-----------------------------------------------------------------------------
                          1591  ;
                          1592  ; Values Represented are in Binary
                          1593  ;
                          1594  ;       MODE - 1=MFM, 0=GCR
                          1595  ;         DN - DRIVE NUMBER
                          1596  ;
                          1597  ; SECTOR SIZE - (MFM ONLY)
                          1598  ;              00 .... 128 BYTE SECTORS
                          1599  ;              01 .... 256 BYTE SECTORS
                          1600  ;              10 .... 512 BYTE SECTORS
                          1601  ;              11 .... 1024 BYTE SECTORS
                          1602  ;
                          1603  ; CONTROLLER STATUS (GCR)
                          1604  ;            000X .... OK
                          1605  ;            0010 .... SECTOR NOT FOUND
                          1606  ;            0011 .... NO SYNC
                          1607  ;            0100 .... DATA BLOCK NOT FOUND
                          1608  ;            0101 .... DATA BLOCK CHECKSUM ERROR
                          1609  ;            0110 .... FORMAT ERROR
                          1610  ;            0111 .... VERIFY ERROR
                          1611  ;            1000 .... WRITE PROTECT ERROR
                          1612  ;            1001 .... HEADER BLOCK CHECKSUM ERROR
                          1613  ;            1010 .... DATA EXTENDS INTO NEXT BLOCK
                          1614  ;            1011 .... DISK ID MISMATCH/ DISK CHANGE
                          1615  ;            1100 .... RESERVED
                          1616  ;            1101 .... RESERVED
                          1617  ;            1110 .... SYNTAX ERROR
                          1618  ;            1111 .... NO DRIVE PRESENT
                          1619  ;
                          1620  ; CONTROLLER STATUS (MFM)
                          1621  ;            000X .... OK
                          1622  ;            0010 .... SECTOR NOT FOUND
                          1623  ;            0011 .... NO ADDRESS MARK
                          1624  ;            0100 .... RESERVED
                          1625  ;            0101 .... DATA CRC ERROR
                          1626  ;            0110 .... FORMAT ERROR
                          1627  ;            0111 .... VERIFY ERROR
                          1628  ;            1000 .... WRITE PROTECT ERROR
                          1629  ;            1001 .... HEADER BLOCK CHECKSUM ERROR
                          1630  ;            1010 .... RESERVED
                          1631  ;            1011 .... DISK CHANGE
                          1632  ;            1100 .... RESERVED
                          1633  ;            1101 .... RESERVED
```

```
Error Addr  Code          Seq    Source statement

                         1634    ;            1110 .... SYNTAX ERROR
                         1635    ;            1111 .... NO DRIVE PRESENT
```

```
Error Addr  Code           Seq    Source statement

                          1637   ;
                          1638   ;EXAMPLE BURST ROUTINES WRITTEN ON C 128
                          1639   ;       *=$1800
                          1640   ;
                          1641   ;  ROUTINE TO READ N-BLOCKS OF DATA
                          1642   ;  COMMAND CHANNEL MUST BE OPEN ON DRIVE
                          1643   ;  OPEN 15,8,15
                          1644   ;  BUFFER AND CMD_BUF, AND CMD_LENGTH MUST BE SETUP
                          1645   ;  PRIOR TO CALLING THIS COMMAND.
                          1646   ;
                          1647   ;serial  = $0a1c        ; fast serial flag
                          1648   ;d2pra   = $dd00
                          1649   ;clkout  = $10
                          1650   ;d1icr   = $dc0d
                          1651   ;d1sdr   = $dc0c
                          1652   ;stat    = $fa
                          1653   ;buffer  = $fb          ; $fb & $fc
                          1654   ;
                          1655   ;       lda  #15        ; logical file number
                          1656   ;       ldx  #8         ; device number
                          1657   ;       ldy  #15        ; secondary address
                          1658   ;       jsr  setlfs     ; setup logical file
                          1659   ;
                          1660   ;       lda  #0         ; noname
                          1661   ;       jsr  setnam     ; setup file name
                          1662   ;
                          1663   ;       jsr  open       ; open logical channel
                          1664   ;
                          1665   ; after the command channel is open subsequent calls should be from 'read'
                          1666   ;
                          1667   ;read    lda  #$00
                          1668   ;       sta  stat       ; clear status
                          1669   ;       lda  serial
                          1670   ;       and  #%10111111 ; clear bit 6 fast serial flag
                          1671   ;       sta  serial
                          1672   ;       ldx  #15
                          1673   ;       jsr  chkout     ; open channel for output
                          1674   ;       ldx  #0
                          1675   ;       ldy  cmd_length ; length of the command
                          1676   ;1$     lda  cmd_buf,x  ; get command
                          1677   ;       jsr  bsout      ; send the command
                          1678   ;       inx
                          1679   ;       dey
                          1680   ;       bne  1$
                          1681   ;
                          1682   ;       jsr  clrchn     ; send eoi
                          1683   ;       bit  serial     ; check speed of drive
                          1684   ;       bvc  error      ; slow serial drive
                          1685   ;
                          1686   ;       sei
```

```
Error Addr  Code           Seq    Source statement

                          1687   ;       bit  d1icr       ; clear interrupt control reg
                          1688   ;       ldx  cmd_buf+5   ; get # of sectors
                          1689   ;       lda  d2pra       ; read serial port
                          1690   ;       eor  #clkout     ; change state of clock
                          1691   ;       sta  d2pra       ; store back
                          1692   ;
                          1693   ;read_itlda  #8
                          1694   ;1$     bit  d1icr       ; wait for byte
                          1695   ;       beq  1$
                          1696   ;
                          1697   ;       lda  d2pra       ; read serial port
                          1698   ;       eor  #clkout     ; change state of clock
                          1699   ;       sta  d2pra       ; store back
                          1700   ;
                          1701   ;       Code to check status byte.
                          1702   ;
                          1703   ;    1) This code will check for mode
                          1704   ;       whether GCR or MFM.
                          1705   ;    2) Verify sector size.
                          1706   ;    3) Check for error, if ok then continue.
                          1707   ;       On error, check error switch if set continue
                          1708   ;       otherwise abort.
                          1709   ;    4) Verify switches
                          1710   ;
                          1711   ;       lda  d1sdr       ; get data from serial data reg
                          1712   ;       sta  stat        ; save status
                          1713   ;       and  #15
                          1714   ;       cmp  #2          ; just check for (3)
                          1715   ;       bcs  error
                          1716   ;
                          1717   ;       ldy  #0          ; even page
                          1718   ;top_rd lda  #8
                          1719   ;1$     bit  d1icr       ; wait for byte
                          1720   ;       beq  1$
                          1721   ;
                          1722   ;       lda  d2pra       ; toggle clock
                          1723   ;       eor  #clkout
                          1724   ;       sta  d2pra
                          1725   ;
                          1726   ;       lda  d1sdr       ; get data
                          1727   ;       sta  (buffer),y  ; save data
                          1728   ;       iny
                          1729   ;       bne  top_rd      ; continue for buffer size
                          1730   ;
                          1731   ;       dex
                          1732   ;       beq  done_read   ; done ?
                          1733   ;
                          1734   ;       inc  buffer+1    ; next buffer
                          1735   ;       jmp  read_it
                          1736   ;
```

```
Error Addr  Code           Seq    Source statement

                          1737    ;done_read
                          1738    ;       clc
                          1739    ;       .byte  $24
                          1740    ;error  sec
                          1741    ;       rts             ; return to sender
                          1742    ;
                          1743    ;cmd_buf .byte 'UO',0,0,0,0,0
                          1744    ;
```

```
Error Addr  Code           Seq    Source statement

                          1846   ;       sta  stat       ; save status
                          1847   ;       jsr  clkhi      ; set clock high
                          1848   ;
                          1849   ;       Code to check status byte.
                          1850   ;
                          1851   ;    1) This code will check for mode
                          1852   ;       whether GCR or MFM.
                          1853   ;    2) Verify sector size.
                          1854   ;    3) Check for error, if ok then continue.
                          1855   ;       On error, check error switch if set continue
                          1856   ;       otherwise abort.
                          1857   ;    4) Verify switches
                          1858   ;
                          1859   ;       lda  stat       ; retrieve status
                          1860   ;       and  #15
                          1861   ;       cmp  #2         ; just chk for error only (3)
                          1862   ;       bcs  error      ; finish
                          1863   ;
                          1864   ;       dex
                          1865   ;       beq  done_wr    ; done ?
                          1866   ;
                          1867   ;       inc  buffer+1   ; next buffer
                          1868   ;       jmp  wrt_it
                          1869   ;
                          1870   ;done_wr
                          1871   ;       cli
                          1872   ;       .byte $24
                          1873   ;error  sec
                          1874   ;       rts
                          1875   ;
                          1876   ;cmd_buf .byte 'U0',2,0,0,0,0
                          1877   ;
                          1878   ; subroutines
                          1879   ;
                          1880   ; NOTE: spout and spin are C128 kernal vectors (refer to C128 users manual)
                          1881   ;
                          1882   ;
                          1883   ;spout  lda  mmureg     ; change serial direction to output
                          1884   ;       ora  #$08
                          1885   ;       sta  mmureg
                          1886   ;       lda  #$7f
                          1887   ;       sta  d1icr      ; no irq's
                          1888   ;       lda  #$00
                          1889   ;       sta  d1timh
                          1890   ;       lda  #$03
                          1891   ;       sta  d1timl     ; low 6 us bit (fastest)
                          1892   ;       lda  d1cra
                          1893   ;       and  #$80       ; keep TOD
                          1894   ;       ora  #$55
                          1895   ;       sta  d1cra      ; setup CRA for output
```

```
Error Addr  Code           Seq    Source statement

                          1896   ;       bit  d1icr      ; clr pending
                          1897   ;       rts
                          1898   ;
                          1899   ;spin   lda  d1cra
                          1900   ;       and  #$80        ; save TOD
                          1901   ;       ora  #$08        ; change fast serial for input
                          1902   ;       sta  d1cra       ; input for 6526
                          1903   ;       lda  mmureg
                          1904   ;       and  #$f7
                          1905   ;       sta  mmureg      ; mmu serial direction in
                          1906   ;       rts
                          1907   ;
                          1908   ;clklo  lda  d2pra       ; set clock low
                          1909   ;       ora  #clkout
                          1910   ;       sta  d2pra
                          1911   ;       rts
                          1912   ;
                          1913   ;clkhi  lda  d2pra       ; set clock high
                          1914   ;       and  #$ff-clkout
                          1915   ;       sta  d2pra
                          1916   ;       rts
                          1917   ;
                          1918   ; ****** END EXAMPLES ******
                          1919   ;
```

```
Error Addr  Code           Seq    Source statement

                          1921   ; (1) FAST READ
                          1922
     8371   8D 024D       1923   fstrd   sta  cmd         ; save command for dos
     8374   85 5F         1924           sta  ctl_cmd     ; save command for stbctt
                          1925
     8376   AD 180D       1926           lda  ifr1        ; disk change
     8379   4A            1927           lsr  a
     837A   90 18         1928           bcc  frd_00      ; br, ok
                          1929
     837C   A2 0B         1930           ldx  #%00001011 ; no channel
     837E   2C            1931           .byte skip2
     837F   A2 4F         1932   ndkrd   ldx  #%01001111 ; no drive
                          1933
     8381   20 85E9       1934   fail    jsr  upinst      ; update dkmode
     8384   20 8581       1935   finbad  jsr  statqy      ; wait for state handshake
     8387   E0 02         1936   final   cpx  #2
     8389   B0 01         1937           bcs  exbad
                          1938
     838B   60            1939           rts
                          1940
     838C   8A            1941   exbad   txa              ; retrieve status
     838D   29 0F         1942           and  #15         ; bits 0-3 only
     838F   A2 00         1943           ldx  #0          ; jobnum
     8391   4C E60A       1944           jmp  error       ; controller error entry
                          1945
                          1946
     8394   20 81CE       1947   frd_00  jsr  spout
                          1948
     8397   24 5E         1949           bit  dkmode      ; gcr or mfm ?
     8399   10 05         1950           bpl  rdgcr
                          1951
                          1952
                          1953   ;       MFM READ
                          1954
     839B   A9 09         1955           lda  #9          ; read cmd
     839D   4C 86E6       1956           jmp prcmd
                          1957
     83A0   20 C63D       1958   rdgcr   jsr  autoi       ; check for auto init
                          1959
     83A3   58            1960   frd_01  cli              ; let controller run
     83A4   A5 3B         1961           lda  switch      ; check for B
     83A6   29 20         1962           and  #%00100000
     83A8   D0 26         1963           bne  bfonly
                          1964
     83AA   AD 0203       1965           lda  cmdbuf+3    ; get track
     83AD   85 06         1966           sta  hdrs
     83AF   AD 0204       1967           lda  cmdbuf+4    ; get sector
     83B2   85 07         1968           sta  hdrs+1
     83B4   A2 00         1969           ldx  #0          ; job #0
     83B6   A5 5F         1970           lda  ctl_cmd     ; read ($80) or fread ($88)
```

```
Error Addr  Code           Seq    Source statement

     83B8   95 00         1971            sta  jobs,x
     83BA   20 865E       1972            jsr  stbctt      ; hit controller hard
                          1973
     83BD   78            1974            sei              ; disable irqs
     83BE   20 85E9       1975            jsr  upinst      ; update controller status
                          1976
     83C1   24 3B         1977            bit  switch      ; check E
     83C3   70 04         1978            bvs  igerr
                          1979
     83C5   E0 02         1980            cpx  #2          ; error
     83C7   B0 B8         1981            bcs  fail
                          1982
     83C9   20 85F9       1983    igerr   jsr  hskrd       ; handshake on state of clkin
                          1984
     83CC   A5 3B         1985            lda  switch      ; check B
     83CE   30 0D         1986            bmi  notran
                          1987
     83D0   A0 00         1988    bfonly  ldy  #0          ; even page
     83D2   B9 0300       1989    frd_02  lda  $0300,y     ; get data
     83D5   85 46         1990            sta  ctl_dat     ; setup data
     83D7   20 85F9       1991            jsr  hskrd       ; handshake on state
     83DA   C8            1992            iny
     83DB   D0 F5         1993            bne  frd_02
                          1994
     83DD   CE 0205       1995    notran  dec  cmdbuf+5    ; any more sectors ?
     83E0   F0 06         1996            beq  exrd
                          1997
     83E2   20 861E       1998            jsr  sektr       ; next sector
     83E5   4C 83A3       1999            jmp  frd_01      ; more to do
                          2000
     83E8   58            2001    exrd    cli
     83E9   4C 85AF       2002            jmp  chksee      ; next track ?
                          2003
```

```
Error Addr  Code           Seq    Source statement

                          2005   ; (2) FAST WRITE
                          2006
     83EC   8D 024D       2007   fstwrt  sta  cmd         ; save command for dos
     83EF   AD 180D       2008           lda  ifr1        ; disk change
     83F2   4A            2009           lsr  a
     83F3   90 0D         2010           bcc  fwrt_0      ; br, ok
                          2011
     83F5   A2 0B         2012           ldx  #%00001011 ; no channel
     83F7   2C            2013           .byte  skip2
     83F8   A2 4F         2014   ndkwrt  ldx  #%01001111 ; no drv 1
     83FA   86 46         2015           stx  ctl_dat     ; save status
     83FC   A5 3B         2016           lda  switch      ; set internal switch
     83FE   09 08         2017           ora  #%00001000
     8400   85 3B         2018           sta  switch
                          2019
     8402   24 5E         2020   fwrt_0  bit  dkmode      ; gcr or mfm ?
     8404   10 05         2021           bpl  wrtgcr
                          2022
                          2023
                          2024   ;       MFM WRITE
                          2025
     8406   A9 0A         2026           lda  #10
     8408   4C 86E6       2027           jmp  prcmd
                          2028
     840B   20 C63D       2029   wrtgcr  jsr  autoi       ; chk for autoi
                          2030
     840E   A5 3B         2031           lda  switch      ; transmision required
     8410   30 29         2032           bmi  notrx
                          2033
     8412   78            2034   fwrt_1  sei              ; no irqs
     8413   A0 00         2035           ldy  #0          ; even page
     8415   AD 1800       2036   fwrt_2  lda  pb          ; debounce
     8418   49 08         2037           eor  #clkout     ; toggle state of clock
     841A   2C 400D       2038           bit  icr         ; clear pending
     841D   8D 1800       2039           sta  pb
     8420   AD 1800       2040   fwrt_4  lda  pb
     8423   10 03         2041           bpl  fwrt_3      ; br, attn not low
                          2042
     8425   20 EA59       2043           jsr  tstatn      ; chk for atn
                          2044
     8428   AD 400D       2045   fwrt_3  lda  icr         ; wait for byte
     842B   29 08         2046           and  #8
     842D   F0 F1         2047           beq  fwrt_4      ; chk for attn low only
                          2048
     842F   AD 400C       2049           lda  sdr         ; get data
     8432   99 0300       2050           sta  $0300,y     ; put away data
     8435   C8            2051           iny
     8436   D0 DD         2052           bne  fwrt_2      ; more ?
                          2053
     8438   20 E9B7       2054           jsr  clkhi       ; release clock
```

```
Error Addr  Code           Seq    Source statement

     843B   58            2055   notrx   cli             ; let controller run
     843C   A5 3B         2056           lda  switch      ; check for buffer transfer only
     843E   29 20         2057           and  #%00100000
     8440   D0 37         2058           bne  nowrt
                          2059
     8442   A5 3B         2060           lda  switch      ; check internal
     8444   29 08         2061           and  #%00001000
     8446   F0 05         2062           beq  notrx0      ; br, ok...
                          2063
     8448   A6 46         2064           ldx  ctl_dat     ; get error
     844A   4C 8381       2065           jmp  fail        ; abort
                          2066
     844D   AD 0203        2067   notrx0  lda  cmdbuf+3   ; get track
     8450   85 06         2068           sta  hdrs
     8452   AD 0204       2069           lda  cmdbuf+4    ; get sector
     8455   85 07         2070           sta  hdrs+1
     8457   A2 00         2071           ldx  #0          ; job #0
     8459   A9 90         2072           lda  #write      ; get write job
     845B   95 00         2073           sta  jobs,x
     845D   20 865E       2074           jsr  stbctt      ; wack controller in the head
                          2075
     8460   78            2076           sei             ; no irqs ok !
     8461   20 81CE       2077           jsr  spout       ; go output
     8464   20 85E9       2078           jsr  upinst      ; update status
     8467   20 85F9       2079           jsr  hskrd       ; send it
     846A   20 86A0       2080           jsr  burst       ; wait for reverse
     846D   20 81B2       2081           jsr  spinp
     8470   58            2082           cli             ; irqs ok now
                          2083
     8471   24 3B         2084           bit  switch      ; check error abort switch
     8473   70 04         2085           bvs  nowrt
                          2086
     8475   E0 02         2087           cpx  #2          ; error on job ?
     8477   B0 0B         2088           bcs  fwrt_5      ; abort ?
                          2089
     8479   CE 0205       2090   nowrt   dec  cmdbuf+5   ; more sectors ?
     847C   F0 09         2091           beq  fwrt_6
                          2092
     847E   20 861E       2093           jsr  sektr       ; increment sector
     8481   4C 8412       2094           jmp  fwrt_1
                          2095
     8484   4C 838C       2096   fwrt_5  jmp  exbad
                          2097
     8487   58            2098   fwrt_6  cli             ; done
     8488   4C 85AF       2099           jmp  chksee      ; next track ?
                          2100
```

```
Error Addr  Code           Seq    Source statement

                          2102   ; (3) FAST SEEK
                          2103
      848B   AD 0202      2104   fstsek  lda  cmdbuf+2   ; check drive number
      848E   29 01        2105           and  #1
      8490   D0 20        2106           bne  3$          ; drive 1 - error
                          2107
      8492   A9 01        2108           lda  #1
      8494   8D 180D      2109           sta  ifr1        ; new disk
      8497   A9 05        2110           lda  #5          ; read address cmd
      8499   20 86E6      2111           jsr  prcmd       ; give it to controller
                          2112
      849C   AE 01B0      2113           ldx  mfmcmd      ; get status
      849F   E0 02        2114           cpx  #2          ; ok ?
      84A1   90 11        2115           bcc  2$
                          2116
      84A3   A2 00        2117           ldx  #0
      84A5   86 5E        2118           stx  dkmode      ; force gcr,drv0,ok
      84A7   A9 B0        2119           lda  #seek       ; seek
      84A9   8D 024D      2120           sta  cmd         ; give command to dos
      84AC   95 00        2121           sta  jobs,x      ; give job to controller
      84AE   20 865E      2122           jsr  stbctt      ; bang the controller
      84B1   2C           2123           .byte  skip2     ; return status
      84B2   A2 4F        2124   3$      ldx  #%01001111 ; no drv 1
      84B4   4C 8381      2125   2$      jmp  fail        ; update status
                          2126
```

```
Error Addr  Code           Seq    Source statement

                          2128   ; (4) BURST FORMAT MFM/GCR
                          2129
      84B7   AD 0202      2130   fstfmt  lda  cmdbuf+2    ; check drive number
      84BA   29 01        2131           and  #1
      84BC   D0 2B        2132           bne  1$
                          2133
      84BE   AD 0203      2134           lda  cmdbuf+3    ; format gcr or mfm ?
      84C1   10 05        2135           bpl  2$
                          2136
                          2137
                          2138   ;       FORMAT IN MFM
                          2139
      84C3   A9 08        2140           lda  #8          ; format disk in mfm
      84C5   4C 86E6      2141           jmp  prcmd
                          2142
      84C8   A9 00        2143   2$      lda  #0
      84CA   85 5E        2144           sta  dkmode      ; force GCR,DRV0
      84CC   85 FF        2145           sta  nodrv       ; drive ok
      84CE   AD 0204      2146           lda  cmdbuf+4
      84D1   85 12        2147           sta  dskid       ; low id
      84D3   AD 0205      2148           lda  cmdbuf+5
      84D6   85 13        2149           sta  dskid+1     ; high id
                          2150
      84D8   20 D307      2151           jsr  clrchn      ; close all channels
      84DB   A9 01        2152           lda  #1
      84DD   85 80        2153           sta  track       ; track one
      84DF   A9 FF        2154           lda  #$ff
      84E1   8D 0298      2155           sta  jobrtn      ; set error recovery
      84E4   20 A989      2156           jsr  ptch42      ; format
                          2157
      84E7   AA           2158           tax              ; save error
      84E8   2C           2159           .byte skip2
      84E9   A2 4F        2160   1$      ldx  #%01001111
      84EB   20 85E9      2161           jsr  upinst      ; update status
      84EE   4C 8387      2162           jmp  final       ; finish up ...
                          2163
```

```
Error Addr  Code           Seq    Source statement

                          2165   ; (5) SOFT INTERLEAVE
                          2166
     84F1   78            2167   cpmint  sei
     84F2   24 3B         2168           bit  switch      ; read ?
     84F4   10 0A         2169           bpl  1$          ; br, write
                          2170
     84F6   20 81CE       2171           jsr  spout       ; serial port output
     84F9   A5 3C         2172           lda  cpmit       ; get current interleave
     84FB   85 46         2173           sta  ctl_dat     ; send it
     84FD   4C 85F9       2174           jmp  hskrd
                          2175
     8500   AE 0274       2176   1$      ldx  cmdsiz
     8503   E0 04         2177           cpx  #4
     8505   B0 0A         2178           bcs  2$
                          2179
     8507   A2 0E         2180           ldx  #%00001110 ; syntax
     8509   20 85E9       2181           jsr  upinst      ; update dkmode
                          2182
     850C   A9 31         2183           lda  #badcmd
     850E   4C C1C8       2184           jmp  cmderr
                          2185
     8511   AD 0203       2186   2$      lda  cmdbuf+3
     8514   85 3C         2187           sta  cpmit       ; save
     8516   60            2188           rts
```

```
Error Addr  Code           Seq    Source statement

                          2190   ; (6) QUERY DISK FORMAT
                          2191
     8517   20 848B       2192   querdk  jsr  fstsek      ; check disk format
     851A   24 5E         2193           bit  dkmode      ; mfm or gcr ?
     851C   10 48         2194           bpl  2$
                          2195
     851E   A9 0D         2196           lda  #13
     8520   20 86E6       2197   *       jsr  prcmd       ; generate sector table
                          2198
     8523   AE 01B0       2199           ldx  mfmcmd      ; status ok ?
     8526   E0 02         2200           cpx  #2
     8528   B0 08         2201           bcs  1$
                          2202
     852A   20 8961       2203           jsr  maxmin      ; determine low/high sector
     852D   20 8986       2204           jsr  fn_int      ; determine interleave
     8530   8A            2205           txa              ; hard interleave
     8531   48            2206           pha              ; save it
                          2207
     8532   78            2208   1$      sei              ; no irq's
     8533   20 81CE       2209           jsr  spout       ; serial port output
     8536   A5 5E         2210           lda  dkmode      ; get status and send it
     8538   85 46         2211           sta  ctl_dat
     853A   20 85F9       2212           jsr  hskrd       ; transmit status
                          2213
     853D   AE 01B0       2214           ldx  mfmcmd      ; was status ok ?
     8540   E0 02         2215           cpx  #2
     8542   B0 23         2216           bcs  3$
                          2217
     8544   A5 97         2218           lda  cpmsek      ; get number of sectors
     8546   85 46         2219           sta  ctl_dat
     8548   20 85F9       2220           jsr  hskrd       ; send it
                          2221
     854B   A5 67         2222           lda  cmd_trk     ; get logical track number
     854D   85 46         2223           sta  ctl_dat
     854F   20 85F9       2224           jsr  hskrd       ; send track
                          2225
     8552   A5 60         2226           lda  minsek
     8554   85 46         2227           sta  ctl_dat     ; send min.
     8556   20 85F9       2228           jsr  hskrd       ; wait for handshake
                          2229
     8559   A5 61         2230           lda  maxsek      ; send max.
     855B   85 46         2231           sta  ctl_dat
     855D   20 85F9       2232           jsr  hskrd       ; wait for handshake
                          2233
     8560   68            2234           pla              ; get interleave
     8561   85 46         2235           sta  ctl_dat     ; send interleave
     8563   20 85F9       2236           jsr  hskrd       ; wait for handshake
                          2237
     8566   60            2238   2$      rts
     8567   68            2239   3$      pla              ; restore stack
```

```
Error Addr  Code           Seq    Source statement

     8568   4C 838C       2240            jmp  exbad
```

```
Error Addr  Code           Seq    Source statement

                          2242   ; (7) QUERY/INIT STATUS
                          2243
     856B   24 3B          2244   inqst   bit  switch     ; read/write op ?
     856D   10 27          2245           bpl  wrstat
                          2246
     856F   24 3B          2247           bit  switch     ; wp latch ?
     8571   50 0E          2248           bvc  statqy
                          2249
     8573   AD 180D        2250           lda  ifr1
     8576   4A             2251           lsr  a
     8577   90 08          2252           bcc  statqy     ; ok, give old status
                          2253
     8579   A5 5E          2254           lda  dkmode
     857B   29 F0          2255           and  #$f0       ; clr old
     857D   09 0B          2256           ora  #$0b       ; disk change
     857F   85 5E          2257           sta  dkmode     ; updated status
                          2258
     8581   78             2259   statqy  sei            ; disable irqs
     8582   20 81CE        2260           jsr  spout      ; serial port out
     8585   A5 5E          2261           lda  dkmode     ; get status
     8587   85 46          2262           sta  ctl_dat
     8589   20 85F9        2263           jsr  hskrd      ; 6526 will send it
                          2264
     858C   A9 00          2265           lda  #0
     858E   8D 026C        2266           sta  erword
                          2267
     8591   20 81B2        2268   exchnl  jsr  spinp      ; serial port input
     8594   58             2269   exchn2  cli
     8595   60             2270           rts
                          2271
     8596   AD 0203        2272   wrstat  lda  cmdbuf+3   ; new value
     8599   85 5E          2273           sta  dkmode
     859B   24 3B          2274           bit  switch     ; wp latch ?
     859D   50 05          2275           bvc  1$
                          2276
     859F   A9 01          2277           lda  #1
     85A1   8D 180D        2278           sta  ifr1       ; new disk
                          2279
     85A4   60             2280   1$      rts
```

```
Error Addr  Code           Seq    Source statement

                          2282   ; (8) BACKUP DISK
                          2283
     85A5   A2 0E         2284   duplc1  ldx  #%00001110
     85A7   20 85E9       2285           jsr  upinst      ; update dkmode
                          2286
     85AA   A9 31         2287           lda  #badcmd
     85AC   4C C1C8       2288           jmp  cmderr
                          2289
```

```
Error Addr  Code           Seq    Source statement

                          2291   ;       SUBROUTINES
                          2292
     85AF   AD 0274       2293   chksee  lda  cmdsiz     ; chk for next track
     85B2   C9 07         2294           cmp  #7
     85B4   90 32         2295           bcc  3$
                          2296
     85B6   A5 06         2297           lda  hdrs       ; where are we ?
     85B8   A8            2298           tay             ; save
     85B9   E9 01         2299           sbc  #1         ; one less
     85BB   0A            2300           asl  a
     85BC   85 64         2301           sta  cur_trk
     85BE   C0 24         2302           cpy  #36
     85C0   08            2303           php             ; save status
     85C1   AC 0206       2304           ldy  cmdbuf+6   ; destination
     85C4   84 22         2305           sty  drvtrk
     85C6   88            2306           dey
     85C7   84 67         2307           sty  cmd_trk    ; initial detent
     85C9   C0 23         2308           cpy  #35
     85CB   6A            2309           ror  a          ; rotate carry into neg
     85CC   28            2310           plp
     85CD   29 80         2311           and  #$80       ; set/clr neg
     85CF   90 0B         2312           bcc  1$         ; br, we are on side zero
                          2313
     85D1   30 12         2314           bmi  2$         ; br, we are on side one & want to goto side one
                          2315
     85D3   18            2316           clc
     85D4   A5 67         2317           lda  cmd_trk    ; setup for pete seke
     85D6   69 23         2318           adc  #35        ; select side
     85D8   85 67         2319           sta  cmd_trk
     85DA   30 09         2320           bmi  2$         ; bra more or less
                          2321
     85DC   10 07         2322   1$      bpl  2$         ; br, we are on side zero & want to goto side zero
                          2323
     85DE   38            2324           sec
     85DF   A5 67         2325           lda  cmd_trk
     85E1   E9 23         2326           sbc  #35        ; select side
     85E3   85 67         2327           sta  cmd_trk
     85E5   4C 87BA       2328   2$      jmp  seke       ; do it
     85E8   60            2329   3$      rts
                          2330
                          2331   ;*******************************************************
                          2332
     85E9   86 46         2333   upinst  stx  ctl_dat
     85EB   A5 5E         2334           lda  dkmode     ; update main status w/ controller status
     85ED   29 F0         2335           and  #%11110000 ; clear old controller status
     85EF   05 46         2336           ora  ctl_dat    ; or in controller status
     85F1   85 5E         2337           sta  dkmode     ; updated
     85F3   85 46         2338           sta  ctl_dat
     85F5   60            2339           rts
                          2340
```

```
Error Addr  Code           Seq    Source statement

                          2341   ;*****************************************************
                          2342
     85F6   20 EA59       2343   hsktst  jsr  tstatn      ; test for atn
     85F9                 2344   hskrd
     85F9   AD 1800       2345   1$      lda  pb          ; debounce
     85FC   CD 1800       2346           cmp  pb
     85FF   D0 F8         2347           bne  1$
                          2348
     8601   29 FF         2349           and  #$ff        ; set/clr neg flag
     8603   30 F1         2350           bmi  hsktst      ; br, attn low
                          2351
     8605   45 37         2352           eor  fastsr      ; wait for state chg
     8607   29 04         2353           and  #4
     8609   F0 EE         2354           beq  1$
                          2355
     860B   A5 46         2356   hsksnd  lda  ctl_dat     ; retrieve data
     860D   8D 400C       2357           sta  sdr         ; send it
     8610   A5 37         2358           lda  fastsr
     8612   49 04         2359           eor  #4          ; change state of clk
     8614   85 37         2360           sta  fastsr
     8616   A9 08         2361           lda  #8
     8618   2C 400D       2362   1$      bit  icr         ; wait transmission time
     861B   F0 FB         2363           beq  1$
                          2364
     861D   60             2365           rts
                          2366
                          2367   ;*****************************************************
                          2368
     861E   AD 0203       2369   sektr   lda  cmdbuf+3    ; get track
     8621   C9 24         2370           cmp  #36         ; side one ?
     8623   90 02         2371           bcc  1$
                          2372
     8625   E9 23         2373           sbc  #35         ; get offset
     8627   AA            2374   1$      tax
     8628   BD 942B       2375           lda  num_sec-1,x
     862B   AA            2376           tax
     862C   CA            2377           dex
     862D   86 46         2378           stx  ctl_dat     ; save
     862F   18             2379           clc
     8630   AD 0204       2380           lda  cmdbuf+4    ; get sector
     8633   65 3C         2381           adc  cpmit       ; add interleave
     8635   C5 46         2382           cmp  ctl_dat     ; less than maxsec ?
     8637   90 0A         2383           bcc  gtsec
     8639   E5 46         2384           sbc  ctl_dat     ; rap around
     863B   F0 04         2385           beq  oksec       ; special case on max
     863D   38             2386           sec
     863E   E9 01         2387           sbc  #1          ; less one
     8640   2C            2388           .byte skip2
     8641   A5 46         2389   oksec   lda  ctl_dat
     8643   8D 0204       2390   gtsec   sta  cmdbuf+4    ; new sector
```

```
Error Addr  Code           Seq    Source statement

     8646   A9 88         2391           lda  #fread      ; exread job
     8648   85 5F         2392           sta  ctl_cmd
     864A   60            2393           rts
                          2394
                          2395   ;*****************************************************
                          2396
     864B   A6 F9         2397   stbctr  ldx  jobnum
     864D   08            2398           php
     864E   58            2399           cli               ; let controller run free
     864F   20 9FB6       2400           jsr  stbctl       ; strobe controller ( hit him hard )
     8652   C9 02         2401           cmp  #2           ; was there an error ?
     8654   90 05         2402           bcc  1$           ; br, nope
                          2403
     8656   20 8683       2404           jsr  stbret       ; let DOS retry
     8659   B5 00         2405           lda  jobs,x       ; get error
                          2406
     865B   AA            2407   1$      tax               ; return status in .x
     865C   28            2408           plp               ; restore status
     865D   60            2409           rts
                          2410
                          2411   ;*****************************************************
                          2412
     865E   A2 00         2413   stbctt  ldx  #0           ; buffer zero
     8660   08            2414           php
     8661   78            2415           sei               ; no irq's during i/o access
     8662   AD 1C00       2416           lda  ledprt
     8665   09 08         2417           ora  #8
     8667   8D 1C00       2418           sta  ledprt       ; led on
     866A   58            2419           cli               ; let controller run free
     866B   20 9FB6       2420           jsr  stbctl       ; strobe controller ( hit him hard )
     866E   C9 02         2421           cmp  #2           ; was there an error ?
     8670   90 03         2422           bcc  1$           ; br, nope
                          2423
     8672   20 8683       2424           jsr  stbret       ; let DOS retry
                          2425
     8675   78            2426   1$      sei
     8676   AD 1C00       2427           lda  ledprt
     8679   29 F7         2428           and  #$ff-8
     867B   8D 1C00       2429           sta  ledprt       ; led off
     867E   B5 00         2430           lda  jobs,x       ; get error return
     8680   AA            2431           tax               ; return status in .x
     8681   28            2432           plp               ; restore original status
     8682   60            2433           rts
                          2434
                          2435   ;*****************************************************
                          2436
     8683   A9 FF         2437   stbret  lda  #$ff         ; micro-step if you can
     8685   8D 0298       2438           sta  jobrtn       ; set error recovery on
     8688   86 F9         2439           stx  jobnum       ; set job #
     868A   AD 0202       2440           lda  cmdbuf+2     ; get command
```

```
Error Addr  Code           Seq    Source statement

     868D   85 5F         2441           sta  ctl_cmd     ; reset command
     868F   8D 024D       2442           sta  cmd         ; set for dos
     8692   9D 025B       2443           sta . lstjob,x   ; set last job
     8695   85 00         2444           sta  jobs        ; give it to the controller
     8697   20 9FB6       2445           jsr  stbctl      ; wait for it
     869A   4C D599       2446           jmp  watjob      ; let dos clean it up
                          2447
                          2448 ;*****************************************************
                          2449
     869D   20 EA59       2450   burtst  jsr  tstatn      ; test for atn
     86A0                 2451   burst
     86A0   AD 1800       2452   1$      lda  pb          ; debounce
     86A3   CD 1800       2453           cmp  pb
     86A6   D0 F8         2454           bne  1$          ; same ?
                          2455
     86A8   29 FF         2456           and  #$ff        ; set/clr neg flag
     86AA   30 F1         2457           bmi  burtst      ; br, attn low
                          2458
     86AC   45 37         2459           eor  fastsr      ; wait for state chg
     86AE   29 04         2460           and  #4
     86B0   F0 EE         2461           beq  1$
                          2462
     86B2   A5 37         2463           lda  fastsr
     86B4   49 04         2464           eor  #4
     86B6   85 37         2465           sta  fastsr      ; state change
     86B8   60            2466           rts
                          2467
                          2468 ;*****************************************************
                          2469           .include  mfmcntrl
```

```
Error Addr  Code           Seq    Source statement

                           2471            .subttl 'mfmcntrl.src'
                           2472
                           2473    ;bit   7   -            test track
                           2474    ;bit   6   -            test sector
                           2475    ;bit   5   -            test write protect switch
                           2476    ;bit   4   -            motor on
                           2477    ;bit   3   -            seek to track
                           2478    ;bit   2   -            wait for motor up to speed
                           2479    ;bit   1   -            logical seek
                           2480    ;bit   0   -            error return
                           2481
     86B9   00 15 00       2482    cmdinf    .byte  $00,$15,$00,$00,$00,$15,$00,$bc
     86BC   00 00 15
     86BF   00 BC
     86C1   34 DE FE       2483              .byte  $34,$de,$fe,$dc,$15,$15,$00
     86C4   DC 15 15
     86C7   00
                           2484
     86C8   89EC           2485    cmdjmp    .word    cmdzer          ; reset
     86CA   89EF           2486              .word    cmdone          ; bump
     86CC   89FD           2487              .word    cmdtwo          ; wp status
     86CE   8A03           2488              .word    cmdthr          ; setup for step
     86D0   8A08           2489              .word    cmdfor          ; reserved
     86D2   8A09           2490              .word    cmdfve          ; read address
     86D4   87BA           2491              .word    seke            ; seek physical
     86D6   8A86           2492              .word    cmdsev          ; format track
     86D8   8C57           2493              .word    cmdeig          ; format disk
     86DA   8D67           2494              .word    cmdnin          ; read sector
     86DC   8DF6           2495              .word    cmdten          ; write sector
     86DE   8EC6           2496              .word    cmdele          ; verify
     86E0   8F18           2497              .word    cmdtwv          ; verify format
     86E2   8F5F           2498              .word    cmdthi          ; create sector table
     86E4   89B3           2499              .word    diskin          ; is the disk home ?
                           2500
            =001E          2501    cmdis     =   *-cmdjmp
```

```
Error Addr  Code           Seq    Source statement

     86E6   78            2503   prcmd   sei
     86E7   48            2504           pha             ; save cmd
     86E8   AA            2505           tax
     86E9   BD 86B9       2506           lda  cmdinf,x   ; get information
     86EC   85 1B         2507           sta  info
     86EE   A5 5E         2508           lda  dkmode
     86F0   09 80         2509           ora  #$80       ; mfm mode
     86F2   85 5E         2510           sta  dkmode
                          2511
     86F4   06 1B         2512           asl  info       ; check track
     86F6   90 05         2513           bcc  prc00
                          2514
     86F8   AD 0203       2515           lda  cmdbuf+3
     86FB   85 67         2516           sta  cmd_trk    ; save track
     86FD   06 1B         2517   prc00   asl  info       ; check sector
     86FF   90 05         2518           bcc  prc01
                          2519
     8701   AD 0204       2520           lda  cmdbuf+4
     8704   85 43         2521           sta  sectr      ; save sector
     8706   06 1B         2522   prc01   asl  info       ; check wp for write jobs
     8708   90 11         2523           bcc  prc02
                          2524
     870A   AD 1C00       2525           lda  dskcnt
     870D   29 10         2526           and  #$10
     870F   D0 0A         2527           bne  prc02
                          2528
     8711   A5 3B         2529           lda  switch     ; set kill job switch
     8713   09 08         2530           ora  #%00001000
     8715   85 3B         2531           sta  switch     ; wp error
     8717   A2 08         2532           ldx  #8
     8719   86 46         2533           stx  ctl_dat
                          2534
     871B   06 1B         2535   prc02   asl  info       ; motor on ?
     871D   90 03         2536           bcc  prc03
                          2537
     871F   20 8794       2538           jsr  stmtr      ; start motor
     8722   06 1B         2539   prc03   asl  info       ; seek ?
     8724   90 03         2540           bcc  prc04      ; seek physical while motor is spinning
                          2541
     8726   20 87BA       2542           jsr  seke       ; seek job
     8729   06 1B         2543   prc04   asl  info       ; wait for motor up to speed ?
     872B   90 03         2544           bcc  prc05
                          2545
     872D   20 87B0       2546           jsr  wtmtr      ; go and wait
                          2547
                          2548   ; disk up to speed
                          2549
     8730   20 8954       2550   prc05   jsr  sel_sid    ; select
     8733   06 1B         2551           asl  info       ; logical seek required ?
     8735   90 03         2552           bcc  prc06
```

```
Error Addr  Code           Seq    Source statement

                           2553
     8737   20 892A        2554            jsr  s_log      ; seek by reading
                           2555
                           2556    ; disk up to speed and on track
                           2557
     873A   A9 00          2558    prc06   lda  #0
     873C   68             2559            pla             ; get jobs
     873D   0A              2560            asl  a
     873E   AA              2561            tax
     873F   BD 86C8         2562            lda  cmdjmp,x   ; do indirect
     8742   85 6F          2563            sta  temp
     8744   BD 86C9         2564            lda  cmdjmp+1,x
     8747   85 70          2565            sta  temp+1
                           2566
     8749   20 8761        2567            jsr  do_the_command
                           2568
     874C   20 F98F        2569            jsr  trnoff     ; motor off...
                           2570
     874F   AE 01B0        2571            ldx  mfmcmd
     8752   E0 02          2572            cpx  #2
     8754   08             2573            php
                           2574
     8755   06 1B          2575            asl  info
     8757   B0 06          2576            bcs  prc07
                           2577
     8759   28             2578            plp
     875A   90 04          2579            bcc  prc08
                           2580
     875C   4C 838C        2581            jmp  exbad
                           2582
     875F   28              2583    prc07   plp             ; restore status
                           2584
     8760   60.             2585    prc08   rts
                           2586
     8761   6C 006F        2587    do_the_command  jmp  (temp)
                           2588
```

```
Error Addr  Code           Seq    Source statement

                           2590
                           2591
                           2592   ;*****************************************************
                           2593
     8764   08             2594   moton   php             ; no irq's
     8765   78             2595           sei
     8766   AD 1C00        2596           lda  dskcnt
     8769   09 04          2597           ora  #4
     876B   8D 1C00        2598           sta  dskcnt
     876E   28             2599           plp             ; retrieve status
     876F   60             2600           rts
                           2601
                           2602   ;*****************************************************
                           2603
     8770   08             2604   motoff  php             ; no irq's
     8771   78             2605           sei
     8772   AD 1C00        2606           lda  dskcnt
     8775   29 FB          2607           and  #$ff-$04
     8777   8D 1C00        2608           sta  dskcnt
     877A   28             2609           plp             ; retrieve status
     877B   60             2610           rts
                           2611
                           2612   ;*****************************************************
                           2613
     877C   08             2614   led_on  php             ; no irq's
     877D   78             2615           sei
     877E   AD 1C00        2616           lda  dskcnt     ; turn on led
     8781   09 08          2617           ora  #8
     8783   8D 1C00        2618           sta  dskcnt
     8786   28             2619           plp             ; retrieve status
     8787   60             2620           rts
                           2621
                           2622   ;*****************************************************
                           2623
     8788   08             2624   led_off php             ; no irq's
     8789   78             2625           sei
     878A   AD 1C00        2626           lda  dskcnt     ; turn led off
     878D   29 F7          2627           and  #$ff-$08
     878F   8D 1C00        2628           sta  dskcnt
     8792   28             2629           plp             ; retrieve status
     8793   60             2630           rts
                           2631
                           2632   ;*****************************************************
                           2633
     8794   08             2634   stmtr   php
     8795   78             2635           sei
     8796   A5 20          2636           lda  drvst
     8798   C9 20          2637           cmp  #$20       ; disk up to speed
     879A   F0 0E          2638           beq  resoff
                           2639
```

```
Error Addr  Code           Seq    Source statement

     879C   AD 0202       2640            lda  cmdbuf+2   ; get drive
     879F   29 01         2641            and  #1
     87A1   85 3E         2642            sta  cdrive     ; set drive
                          2643
     87A3   20 8764       2644            jsr  moton      ; turn on motor
     87A6   A9 A0         2645            lda  #$a0       ; accelerating
     87A8   85 20         2646            sta  drvst      ; drv not ready
     87AA   A9 32         2647   resoff   lda  #50        ; lo
     87AC   85 48         2648            sta  acltim
     87AE   28            2649            plp
     87AF   60            2650            rts
                          2651
                          2652   ;*****************************************************
                          2653
     87B0   08            2654   wtmtr    php             ; wait for motor up to speed
     87B1   58            2655            cli
     87B2   A5 20         2656   wtmtr1   lda  drvst
     87B4   C9 20         2657            cmp  #$20       ; drv ready ?
     87B6   D0 FA         2658            bne  wtmtr1
                          2659
     87B8   28            2660            plp
     87B9   60            2661            rts
                          2662
                          2663   ;*****************************************************
                          2664
     87BA   08            2665   seke     php             ; wait for not busy
     87BB   58            2666            cli             ; enable irqs for motor on timing
     87BC   A5 67         2667            lda  cmd_trk
     87BE   0A            2668            asl  a          ; half steps
     87BF   C5 64         2669            cmp  cur_trk    ; correct track ?
     87C1   F0 1A         2670            beq  sekrtn     ; br, yep
                          2671
     87C3   A5 67         2672   seklop   lda  cmd_trk    ; where
     87C5   0A            2673            asl  a
     87C6   C5 64         2674            cmp  cur_trk
     87C8   F0 0E         2675            beq  sekdun
                          2676
     87CA   B0 06         2677            bcs  sekup      ; hi or lo ?
                          2678
     87CC   20 87E7       2679   sekdn    jsr  stout      ; step out
     87CF   4C 87C3       2680            jmp  seklop     ; next
                          2681
     87D2   20 87DF       2682   sekup    jsr  stin       ; step in
     87D5   4C 87C3       2683            jmp  seklop     ; next also
                          2684
     87D8   A0 10         2685   sekdun   ldy  #$10       ; settle
     87DA   20 8829       2686            jsr  xms        ; delay
                          2687
     87DD   28            2688   sekrtn   plp
     87DE   60            2689            rts
```

```
Error Addr  Code           Seq    Source statement

                          2690
                          2691   ;*****************************************************
                          2692
     87DF   A5 64         2693   stin    lda  cur_trk     ; get current valuestep
     87E1   18            2694           clc
     87E2   69 01         2695           adc  #1
     87E4   4C 8814       2696           jmp  stpfin      ; half step out
                          2697
     87E7   A0 63         2698   stout   ldy  #99         ; wait for trk_00
     87E9   AD 180F       2699   1$      lda  pota1       ; check for trk_00
     87EC   6A            2700           ror  a           ; rotate into carry
     87ED   08            2701           php              ; save carry
     87EE   AD 180F       2702           lda  pota1       ; debounce it
     87F1   6A            2703           ror  a           ; carry <=
     87F2   6A            2704           ror  a           ; bit 7 <=
     87F3   28            2705           plp              ; retrieve carry
     87F4   29 80         2706           and  #$80        ; set/clear sign bit
     87F6   90 04         2707           bcc  3$          ; on track zero
                          2708
     87F8   10 15         2709           bpl  2$          ; carry set(off) & sign(on) exit
                          2710
     87FA   30 02         2711           bmi  4$          ; bra
                          2712
     87FC   30 11         2713   3$      bmi  2$          ; carry clear(on) & sign(off) exit
                          2714
     87FE   88            2715   4$      dey
     87FF   D0 E8         2716           bne  1$          ; continue
                          2717
     8801   B0 0C         2718           bcs  2$          ; br, not track 00 ?
                          2719
     8803   AD 1C00       2720           lda  dskcnt      ; phase 0
     8806   29 03         2721           and  #3
     8808   D0 05         2722           bne  2$
                          2723
     880A   A9 00         2724           lda  #0          ; trk zero
     880C   85 64         2725           sta  cur_trk     ; current trk = 40*2/0
     880E   60            2726           rts
                          2727
     880F   A5.64         2728   2$      lda  cur_trk     ; get current half step
     8811   38            2729           sec
     8812   E9 01         2730           sbc #1
                          2731
     8814   85 64         2732   stpfin  sta  cur_trk     ; save
     8816   29 03         2733           and  #3          ; strip unused
     8818   85 6F         2734           sta  temp
     881A   08            2735           php              ; no irq's
     881B   78            2736           sei
     881C   AD 1C00       2737           lda  dskcnt
     881F   29 FC         2738           and  #$fc        ; strip step bits
     8821   05 6F         2739           ora  temp
```

```
Error Addr  Code          Seq   Source statement

     8823   8D 1C00      2740          sta  dskcnt
     8826   28           2741          plp             ; retrieve status
     8827   A0 06        2742          ldy  #$06       ; 6 ms step time, 16 ms settle
                         2743
     8829   20 8830      2744  xms     jsr  onems      ; delay 1 ms
     882C   88           2745          dey
     882D   D0 FA         2746          bne  xms
                         2747
     882F   60           2748          rts
                         2749
                         2750
                         2751  ;*****************************************************
                         2752
     8830   A2 02        2753  onems   ldx  #2
     8832   A9 00        2754          lda  #0
     8834   69 01        2755  onems1  adc  #1
     8836   D0 FC        2756          bne  onems1
                         2757
     8838   CA           2758          dex
     8839   D0 F9        2759          bne  onems1
                         2760
     883B   60           2761          rts
                         2762
                         2763
                         2764  ;*****************************************************
                         2765
                         2766
     883C                2767+ cvstat  WDTEST          ; chk address
            =0000        2768A         .ife <*!.$03    ; lower two bits cannot be zero
     883C   EA           2769A         nop             ; fill address error
                         2770A         .endif
     883D   AD 2000       2771          lda  wdstat     ; retrieve status
     8840   4A           2772          lsr  a
     8841   4A           2773          lsr  a
     8842   4A           2774          lsr  a
     8843   29 03        2775          and  #3         ; check status
     8845   AA           2776          tax
     8846   BD 8A82      2777          lda  mfmer,x
     8849   8D 01B0      2778          sta  mfmcmd
     884C   AA           2779          tax
     884D   60           2780          rts
                         2781
                         2782  ;*****************************************************
                         2783
     884E   48           2784  strtwd  pha             ; save accum
     884F   20 877C       2785          jsr  led_on     ; act led on
     8852   68           2786          pla
     8853                2787+         WDTEST          ; chk address
            =0003        2788A         .ife <*!.$03    ; lower two bits cannot be zero
                         2789A         nop             ; fill address error
```

```
Error Addr  Code           Seq    Source statement

                          2790A           .endif
     8853   8D 2000       2791            sta  wdstat      ; send cmd to wd 1770
     8856   A9 01         2792            lda  #1          ; wait for busy
     8858                  2793+           WDTEST           ; chk address
            =0000          2794A           .ife <*!.$03     ; lower two bits cannot be zero
     8858   EA             2795A           nop              ; fill address error
                           2796A           .endif
     8859   2C 2000       2797   1$       bit  wdstat
     885C   F0 FB         2798            beq  1$
                          2799
     885E   4C A47E       2800            jmp  jslower     ; WD1770 bug ????
                          2801
                          2802
                          2803   ;*****************************************************
                          2804
     8861   20 8788       2805   waitdn   jsr  led_off     ; act led off
     8864   A9 01         2806            lda  #1          ; wait for unbusy
     8866                  2807+           WDTEST           ; chk address
            =0002          2808A           .ife <*!.$03     ; lower two bits cannot be zero
                           2809A           nop              ; fill address error
                           2810A           .endif
     8866   2C 2000       2811   1$       bit  wdstat
     8869   D0 FB         2812            bne  1$
                          2813
     886B   60             2814            rts
                          2815
```

```
Error Addr  Code           Seq    Source statement

                          2817
                          2818   ;*****************************************************
                          2819
     886C   A5 60         2820   sectnx  lda  minsek
     886E   38            2821           sec
     886F   E9 01         2822           sbc  #1
     8871   85 46         2823           sta  ctl_dat     ; save min sector - 1
     8873   AD 0204       2824           lda  cmdbuf+4    ; get original sector
     8876   18            2825           clc
     8877   65 3C         2826           adc  cpmit       ; next sector
     8879   C5 61         2827           cmp  maxsek
     887B   F0 07         2828           beq  2$          ; equal to or
                          2829
     887D   90 05         2830           bcc  2$          ;          less than
                          2831
     887F   E5 61         2832           sbc  maxsek      ; rap around
     8881   18            2833           clc
     8882   65 46         2834           adc  ctl_dat     ; add min now
     8884   8D 0204       2835   2$      sta  cmdbuf+4    ; next sector for controller
     8887   60            2836           rts
                          2837
```

```
Error Addr  Code           Seq    Source statement

                           2839
                           2840   ;*****************************************************
                           2841
     8888   A0 00          2842   sectcv  ldy  #0
     888A   A2 00          2843           ldx  #0
     888C   AD 0203        2844           lda  cmdbuf+3    ; start at sector x
     888F   29 3F          2845           and  #$3f        ; delete mode bit & table bit
     8891   8D 0203        2846           sta  cmdbuf+3
     8894   85 60          2847           sta  minsek      ; new min
     8896   48             2848           pha              ; save ss
     8897   AD 0207        2849           lda  cmdbuf+7    ; save ns
     889A   48             2850           pha
     889B   EE 0204        2851           inc  cmdbuf+4    ; interleave + 1
                           2852
     889E   AD 0203        2853   sectop  lda  cmdbuf+3    ; ss
     88A1   99 020B        2854           sta  cmdbuf+11,y ; table goes to cmdbuf+42
     88A4   EE 0203        2855           inc  cmdbuf+3    ; increment ss
     88A7   E8             2856           inx
     88A8   98             2857           tya              ; do addition
     88A9   18             2858           clc
     88AA   6D 0204        2859           adc  cmdbuf+4    ; add interleave
     88AD   A8             2860           tay
     88AE   C0 20          2861           cpy  #32         ; no more room
     88B0   B0 0C          2862           bcs  bd_int
                           2863
     88B2   CC 0207        2864           cpy  cmdbuf+7    ; ns
     88B5   90 1A          2865           bcc  secles      ; < than
                           2866
     88B7   D0 12          2867           bne  seclmo      ; can't be equal until the end
                           2868
     88B9   EC 0207        2869           cpx  cmdbuf+7    ; ns end ?
     88BC   F0 0D          2870           beq  seclmo
                           2871
     88BE   CE 0204        2872   bd_int  dec  cmdbuf+4    ; restore
     88C1   68             2873           pla
     88C2   8D 0207        2874           sta  cmdbuf+7    ; restore ns
     88C5   68             2875           pla
     88C6   8D 0203        2876           sta  cmdbuf+3    ; restore ss
                           2877
     88C9   38             2878           sec
     88CA   60             2879           rts              ; bad exit
                           2880
     88CB   98             2881   seclmo  tya
     88CC   38             2882           sec
     88CD   ED 0207        2883           sbc  cmdbuf+7    ; ns
     88D0   A8             2884           tay
                           2885
     88D1   EC 0207        2886   secles  cpx  cmdbuf+7    ; ns
     88D4   D0 C8          2887           bne  sectop
                           2888
```

```
Error Addr  Code           Seq    Source statement

     88D6   86 97         2889            stx  cpmsek      ; ns
     88D8   CA            2890            dex
     88D9   8A            2891            txa
     88DA   18            2892            clc
     88DB   65 60         2893            adc  minsek      ; find max
     88DD   85 61         2894            sta  maxsek      ; new max
     88DF   C5 60         2895            cmp  minsek      ; better be less than max
     88E1   90 DB         2896            bcc  bd_int      ; bad starting sector
                          2897
     88E3   68            2898            pla
     88E4   8D 0207       2899            sta  cmdbuf+7    ; restore ns
     88E7   68            2900            pla
     88E8   8D 0203       2901            sta  cmdbuf+3    ; restore ss
     88EB   CE 0204       2902            dec  cmdbuf+4    ; restore
                          2903
     88EE   18            2904            clc              ; ok exit
     88EF   60            2905            rts
                          2906
                          2907
                          2908    ;*****************************************************
                          2909
                          2910
     88F0   AD 01B0       2911    verfmt  lda  mfmcmd
     88F3   48            2912            pha              ; save current logical track
                          2913
     88F4   A0 00         2914            ldy  #0
     88F6   84 24         2915            sty  mfmhdr      ; init index into sector table
     88F8   A4 24         2916    ver_fm  ldy  mfmhdr      ; get current index
     88FA   B9 020B       2917            lda  cmdbuf+11,y
     88FD   8D 2002       2918            sta  wdsec       ; setup sector for wd 1770
     8900   20 8F18       2919            jsr  cmdtwv      ; verify
     8903   AE 01B0       2920            ldx  mfmcmd      ; get error
     8906   E0 02         2921            cpx  #2
     8908   B0 0B         2922            bcs  ver_00
                          2923
     890A   E6 24         2924            inc  mfmhdr      ; next entry in sector table
     890C   A4 24         2925            ldy  mfmhdr
     890E   CC 0207       2926            cpy  cmdbuf+7    ; check with ns
     8911   D0 E5         2927            bne  ver_fm      ; done ?
                          2928
     8913   18            2929            clc
     8914   24            2930            .byte  skip1
     8915   38            2931    ver_00  sec
     8916   68            2932            pla
     8917   8D 01B0       2933            sta  mfmcmd      ; restore track address
     891A   60            2934    no_trk  rts
                          2935
                          2936
                          2937    ;*****************************************************
                          2938
```

```
Error Addr  Code           Seq    Source statement

     891B   AD 0274       2939   chcsee  lda  cmdsiz
     891E   C9 07         2940           cmp  #7          ; next track parameter given ?
     8920   90 F8         2941           bcc  no_trk
                          2942
     8922   AD 0206       2943           lda  cmdbuf+6    ; get nt
     8925   85 67         2944           sta  cmd_trk
     8927   4C 87BA       2945           jmp  seke
                          2946
                          2947
                          2948   ;*****************************************************
                          2949
                          2950
     892A   AD 01B0       2951   s_log   lda  mfmcmd      ; save command
     892D   48            2952           pha
     892E   20 8A27       2953           jsr  cmdfiv      ; where are we?
     8931   AE 01B0       2954           ldx  mfmcmd      ; status ok?
     8934   E0 02         2955           cpx  #2
     8936   90 0D         2956           bcc  1$          ; see if we are there...
                          2957
     8938   20 89EF       2958           jsr  cmdone      ; restore to trk_00
     893B   20 8A27       2959           jsr  cmdfiv      ; where are we.. better be track 00
     893E   AE 01B0       2960           ldx  mfmcmd      ; status ok?
     8941   E0 02         2961           cpx  #2
     8943   B0 0A         2962           bcs  2$          ; on error leave on track 00
                          2963
     8945   A5 67         2964   1$      lda  cmd_trk     ; we there yet ?
     8947   0A            2965           asl  a
     8948   C5 64         2966           cmp  cur_trk     ; are we on ?
     894A   F0 03         2967           beq  2$
                          2968
     894C   20 87BA       2969           jsr  seke        ; find it...should be there now...hope?
                          2970
     894F   68            2971   2$      pla
     8950   8D 01B0       2972           sta  mfmcmd      ; restore command
     8953   60            2973           rts
                          2974
                          2975
                          2976   ;*****************************************************
                          2977
                          2978
     8954   08            2979   sel_sid php              ; no irq's
     8955   78            2980           sei
     8956   A5 3B         2981           lda  switch
     8958   29 10         2982           and  #%00010000  ; check side
     895A   C9 10         2983           cmp  #%00010000  ; set/clr carry
     895C   20 93F3       2984           jsr  set_side    ; set h/w
     895F   28            2985           plp              ; retrieve status
     8960   60            2986           rts
                          2987
                          2988   ;*****************************************************
```

```
Error Addr  Code           Seq    Source statement

                          2989
      8961  A4 97         2990   maxmin  ldy  cpmsek      ; get ns
      8963  88            2991           dey              ; one less
      8964  A9 FF         2992           lda  #255        ; as small as min can get
      8966  D9 020B       2993   minim   cmp  cmdbuf+11,y
      8969  90 03         2994           bcc  no_min      ; br, no change
                          2995
      896B  B9 020B       2996           lda  cmdbuf+11,y
      896E  88            2997   no_min  dey
      896F  10 F5         2998           bpl  minim
                          2999
      8971  85 60         3000           sta  minsek      ; save min
                          3001
      8973  A4 97         3002           ldy  cpmsek
      8975  88            3003           dey              ; one less
      8976  A9 00         3004           lda  #0          ; as small as max can get
      8978  D9 020B       3005   maxim   cmp  cmdbuf+11,y
      897B  B0 03         3006           bcs  no_max      ; br, no change
                          3007
      897D  B9 020B       3008           lda  cmdbuf+11,y
      8980  88            3009   no_max  dey
      8981  10 F5         3010           bpl  maxim
                          3011
      8983  85 61         3012           sta  maxsek      ; save max
      8985  60            3013           rts
                          3014
                          3015   ;*****************************************************
                          3016
      8986  A6 97         3017   fn_int  ldx  cpmsek      ; location of min
      8988  A0 00         3018           ldy  #0          ; start in beginning
      898A  B9 020B       3019   fn_000  lda  cmdbuf+11,y
      898D  C5 60         3020           cmp  minsek
      898F  F0 05         3021           beq  fn_001      ; min ?
                          3022
      8991  C8            3023           iny
      8992  C4 97         3024           cpy  cpmsek
      8994  D0 F4         3025           bne  fn_000
                          3026
      8996  84 5F         3027   fn_001  sty  ctl_cmd     ; save index
      8998  A5 60         3028           lda  minsek      ; get min
      899A  18            3029           clc
      899B  69 01         3030           adc  #1          ; add one to it
      899D  85 46         3031           sta  ctl_dat     ; save min + 1
                          3032
      899F  A2 FF         3033           ldx  #255        ; interleave at zero
      89A1  B9 020B       3034   fnlook  lda  cmdbuf+11,y
      89A4  C5 46         3035           cmp  ctl_dat     ; find min + 1 ?
      89A6  F0 0A         3036           beq  fn_002
                          3037
      89A8  E8            3038           inx              ; cpmit = cpmit + 1
```

```
Error Addr  Code           Seq    Source statement

                          3097            .subttl 'mfmsubr.src'
                          3098
     89EC   4C EAAO       3099    cmdzer  jmp  dskint      ; reset
                          3100
```

```
Error Addr  Code           Seq    Source statement

     89EF   A9 B4         3102   cmdone  lda  #180
     89F1   85 64         3103           sta  cur_trk
                          3104
     89F3   A9 00         3105           lda  #0          ; side zero
     89F5   8D 2001       3106           sta  wdtrk       ; init to trk zero
     89F8   85 67         3107           sta  cmd_trk     ; find zero
     89FA   4C 87BA       3108           jmp  seke
                          3109
```

```
Error Addr  Code           Seq    Source statement

     89A9   C8             3039            iny
     89AA   C4 97          3040            cpy  cpmsek      ; index here ?
     89AC   DO F3          3041            bne  fnlook
                           3042
     89AE   AO OO          3043            ldy  #0          ; rap around
     89BO   FO EF          3044            beq  fnlook      ; bra
                           3045
     89B2   60             3046   fn_002   rts             ; return with hard interleave in .x
                           3047
                           3048
                           3049   ;*****************************************************
                           3050
                           3051
     89B3   A5 6F          3052   diskin   lda  temp
     89B5   48             3053            pha              ; save temp
     89B6   O8             3054            php              ; save status
     89B7   78             3055            sei              ; no irq's
     89B8   AD 2001        3056            lda  wdtrk       ; where are we ?
     89BB   8D 2003        3057            sta  wddat       ; here we are..
     89BE   A9 18          3058            lda  #$18
     89CO   20 884E        3059            jsr  strtwd      ; seek cmd
     89C3   20 8861        3060            jsr  waitdn      ; fin ?
     89C6   A2 OO          3061            ldx  #0
     89C8   AO 80          3062            ldy  #128        ; wait a little more than a revolution
     89CA                  3063+           WDTEST           ; chk address
            =0002          3064A           .ife <*!.$03     ; lower two bits cannot be zero
                           3065A           nop              ; fill address error
                           3066A           .endif
     89CA   AD 2000        3067            lda  wdstat      ; get start status
     89CD   29 O2          3068            and  #2
     89CF   85 6F          3069            sta  temp        ; save current status
     89D1                  3070+           WDTEST           ; chk address
            =0001          3071A           .ife <*!.$03     ; lower two bits cannot be zero
                           3072A           nop              ; fill address error
                           3073A           .endif
     89D1   AD 2000        3074   1$       lda  wdstat      ; get current status
     89D4   29 O2          3075            and  #2
     89D6   C5 6F          3076            cmp  temp        ; same go on
     89D8   FO O4          3077            beq  2$
                           3078
     89DA   28             3079            plp              ; retrieve status
     89DB   4C 89E7        3080            jmp  4$          ; finish up ok disk inserted
                           3081
     89DE   CA             3082   2$       dex
     89DF   DO FO          3083            bne  1$
                           3084
     89E1   88             3085            dey
     89E2   DO ED          3086            bne  1$          ; timout ?
                           3087
     89E4   28             3088            plp              ; retrieve status
```

```
Error Addr  Code           Seq    Source statement

     89E5   38            3089           sec              ; exit no index sensor toggling
     89E6   24            3090           .byte skip1
     89E7   18            3091    4$     clc              ; ok
     89E8   68            3092           pla
     89E9   85 6F         3093           sta  temp        ; restore temp
     89EB   60            3094           rts
                          3095           .include  mfmsubr
```

```
Error Addr  Code           Seq    Source statement

     89FD   AD 1C00       3111   cmdtwo  lda  dskcnt     ; return status of wp
     8A00   29 10          3112           and  #$10
     8A02   60             3113           rts
                           3114
```

```
Error Addr  Code           Seq    Source statement

     8A03   84 67         3116   cmdthr  sty  cmd_trk
     8A05   86 64         3117           stx  cur_trk
     8A07   60            3118           rts
                          3119
```

```
Error Addr  Code           Seq    Source statement

     8A08   60            3121   cmdfor  rts             ; reserved
                          3122
```

```
Error Addr  Code           Seq    Source statement

                          3124    ;read address routine
                          3125
                          3126    ;  0       1       2           3          4     5 + mfmhdr
                          3127    ;  ^       ^       ^           ^          ^     ^
                          3128    ; track  side#  sector  sector_length  crc   crc
                          3129
                          3130    ; track          => cur-trk
                          3131    ; sector size    => mfmsiz_hi
                          3132    ; status         => mfmsiz_lo
                          3133    ; sector size    => dkmode
                          3134
     8A09   20 89EF       3135    cmdfve  jsr  cmdone      ; restore
     8A0C   20 89B3       3136            jsr  diskin      ; is there a disk in the drive ?
     8A0F   B0 0F         3137            bcs  1$
                          3138
     8A11   20 8A27       3139            jsr  cmdfiv      ; seek
     8A14   BD 8A7E       3140            lda  nsectk,x    ; get # of sectors
     8A17   85 97         3141            sta  cpmsek      ; set def.
     8A19   85 61         3142            sta  maxsek      ; store max sector number....def, ok
     8A1B   A9 01         3143            lda  #1
     8A1D   85 60         3144            sta  minsek      ; let query decide otherwise....def, ok
     8A1F   60            3145            rts
                          3146
     8A20   A9 0D         3147    1$      lda  #$0d        ; no disk
     8A22   8D 01B0       3148            sta  mfmcmd      ; error
     8A25   D0 3E         3149            bne  abrsk       ; abort
                          3150
     8A27   A9 00         3151    cmdfiv  lda  #0
     8A29   8D 0271       3152            sta  mfmsiz_lo   ; clear sector size
     8A2C   85 44         3153            sta  mfmsiz_hi   ; *
     8A2E   A9 C8         3154            lda  #$c8        ; read address
     8A30   20 884E       3155            jsr  strtwd      ; start cmd
                          3156
                          3157    ; store address away
                          3158
     8A33   A2 00         3159            ldx  #0
     8A35   A0 06         3160            ldy  #6
     8A37                 3161+           WDTEST           ; chk address
            =0003         3162A           .ife <*!.$03     ; lower two bits cannot be zero
                          3163A           nop              ; fill address error
                          3164A           .endif
     8A37   AD 2000       3165    mseek1  lda  wdstat
     8A3A   29 03         3166            and  #3
     8A3C   4A            3167            lsr  a
     8A3D   90 0B         3168            bcc  mrcnf       ; no address mark found
     8A3F   F0 F6         3169            beq  mseek1
                          3170
     8A41   AD 2003       3171            lda  wddat       ; get data
     8A44   95 24         3172            sta  mfmhdr,x    ; data in .a
     8A46   E8            3173            inx
```

```
Error Addr  Code           Seq    Source statement

     8A47   88             3174            dey
     8A48   D0 ED          3175            bne  mseek1
                           3176
     8A4A   20 8861        3177   mrcnf    jsr  waitdn      ; wait for not busy
                           3178
                           3179   ; wd 1770 status returns
                           3180
                           3181   ; motor wp spin/rec recnofnd crc lstdat  datreq busy
                           3182   ;   s7  s6    s5       s4     s3   s2      s1    s0
                           3183
     8A4D   20 883C        3184            jsr  cvstat      ; get status
                           3185
     8A50   A5 24          3186   mseek3   lda  mfmhdr      ; read track address
     8A52   0A             3187            asl  a           ; * 2
     8A53   85 64          3188            sta  cur_trk
                           3189
     8A55   A5 27          3190            lda  mfmhdr+3    ; get sector size
     8A57   29 03          3191   consek   and  #3          ; clear remaining
     8A59   AA             3192            tax
     8A5A   BD 8A72        3193            lda  sectlo,x
     8A5D   8D 0271        3194            sta  mfmsiz_lo   ; sector size low
     8A60   BD 8A76        3195            lda  secthi,x
     8A63   85 44          3196            sta  mfmsiz_hi   ; sector size high
                           3197
     8A65   A5 5E          3198   abrsk    lda  dkmode      ; set sector size in
     8A67   29 80          3199            and  #%10000000  ; clear all but mode bit
     8A69   0D 01B0        3200            ora  mfmcmd      ; set status
     8A6C   1D 8A7A        3201            ora  seckzz,x    ; shifted sector size
     8A6F   85 5E          3202            sta  dkmode
     8A71   60             3203            rts
                           3204
                           3205   ; sector size low for mfm 128,256,512,1024 byte sectors
     8A72   7F FF FF       3206   sectlo    .byte  127,255,255,255
     8A75   FF
                           3207
                           3208   ; sector size high for mfm 128,256,512,1024 byte sectors
     8A76   01 01 02       3209   secthi    .byte  1,1,2,4
     8A79   04
                           3210
                           3211   ; shifted sector size for mfm 128,256,512,1024 byte sectors
     8A7A   00 10 20       3212   seckzz    .byte  $00,$10,$20,$30
     8A7D   30
                           3213
                           3214   ; max sector #'s for mfm 128, 256, 512, 1024 byte sectors
     8A7E   1A 10 09       3215   nsectk    .byte  26, 16, 9, 5
     8A81   05
                           3216
                           3217   ; ok, checksum-error, sector-not-found, no-address-mark
     8A82   01 09 02       3218   mfmer     .byte  1,9,2,3
     8A85   03
```

```
Error Addr  Code          Seq    Source statement

                         3219           .include  mfmsubr1
```

```
Error Addr  Code           Seq    Source statement

                          3221            .subttl 'mfmsubr1.src'
                          3222
                          3223    ; format track
                          3224
     8A86   A9 F8         3225    cmdsev  lda  #$f8       ; write track
     8A88   20 884E       3226            jsr  strtwd     ; send command
                          3227
     8A8B   24 3B         3228            bit  switch     ; system 34 / iso standard switch
     8A8D   50 62         3229            bvc  no_ind     ; write index ?
                          3230
     8A8F   A2 50         3231            ldx  #80
     8A91                 3232+           WDTEST          ; chk address
            =0001         3233A           .ife <*!.$03    ; lower two bits cannot be zero
                          3234A           nop             ; fill address error
                          3235A           .endif
     8A91   AD 2000       3236    2$      lda  wdstat
     8A94   29 03         3237            and  #3
     8A96   4A            3238            lsr  a
     8A97   90 60         3239            bcc  v6
     8A99   F0 F6         3240            beq  2$
                          3241
     8A9B   A9 4E         3242            lda  #$4e
     8A9D   8D 2003       3243            sta  wddat      ; give him the data
     8AA0   CA            3244            dex
     8AA1   D0 EE         3245            bne  2$
                          3246
     8AA3   A2 0C         3247            ldx  #12
     8AA5                 3248+           WDTEST          ; chk address
            =0001         3249A           .ife <*!.$03    ; lower two bits cannot be zero
                          3250A           nop             ; fill address error
                          3251A           .endif
     8AA5   AD 2000       3252    3$      lda  wdstat
     8AA8   29 03         3253            and  #3
     8AAA   4A            3254            lsr  a
     8AAB   90 4C         3255            bcc  v6
     8AAD   F0 F6         3256            beq  3$
                          3257
     8AAF   A9 00         3258            lda  #0
     8AB1   8D 2003       3259            sta  wddat
     8AB4   CA            3260            dex
     8AB5   D0 EE         3261            bne  3$
                          3262
     8AB7   A2 03         3263            ldx  #3
     8AB9                 3264+           WDTEST          ; chk address
            =0001         3265A           .ife <*!.$03    ; lower two bits cannot be zero
                          3266A           nop             ; fill address error
                          3267A           .endif
     8AB9   AD 2000       3268    4$      lda  wdstat
     8ABC   29 03         3269            and  #3
     8ABE   4A            3270            lsr  a
```

```
Error Addr  Code           Seq    Source statement

     8ABF   90 38         3271           bcc  v6
     8AC1   F0 F6         3272           beq  4$
                          3273
     8AC3   A9 F6         3274           lda  #$f6
     8AC5   8D 2003       3275           sta  wddat
     8AC8   CA            3276           dex
     8AC9   D0 EE         3277           bne  4$
                          3278
     8ACB                 3279+          WDTEST          ; chk address
            =0003         3280A          .ife <*!.$03    ; lower two bits cannot be zero
                          3281A          nop             ; fill address error
                          3282A          .endif
     8ACB   AD 2000       3283   5$      lda  wdstat
     8ACE   29 03         3284           and  #3
     8AD0   4A            3285           lsr  a
     8AD1   90 26         3286           bcc  v6
     8AD3   F0 F6         3287           beq  5$
                          3288
     8AD5   A9 FC         3289           lda  #$fc
     8AD7   8D 2003       3290           sta  wddat
                          3291
                          3292
     8ADA   A2 32         3293           ldx  #50
     8ADC                 3294+          WDTEST          ; chk address
            =0000         3295A          .ife <*!.$03    ; lower two bits cannot be zero
     8ADC   EA            3296A          nop             ; fill address error
                          3297A          .endif
     8ADD   AD 2000       3298   6$      lda  wdstat
     8AE0   29 03         3299           and  #3
     8AE2   4A            3300           lsr  a
     8AE3   90 14         3301           bcc  v6
     8AE5   F0 F6         3302           beq  6$
                          3303
     8AE7   A9 4E         3304           lda  #$4e
     8AE9   8D 2003       3305           sta  wddat      ; give him the data
     8AEC   CA            3306           dex
     8AED   D0 EE         3307           bne  6$
                          3308
     8AEF   F0 14         3309           beq  inner      ; bra... done...
                          3310
                          3311
     8AF1   A2 3C         3312   no_ind  ldx  #60
     8AF3                 3313+          WDTEST          ; chk address
            =0003         3314A          .ife <*!.$03    ; lower two bits cannot be zero
                          3315A          nop             ; fill address error
                          3316A          .endif
     8AF3   AD 2000       3317   cmd7    lda  wdstat
     8AF6   29 03         3318           and  #3
     8AF8   4A            3319           lsr  a
     8AF9   90 28         3320   v6      bcc  v1
```

```
Error Addr  Code           Seq    Source statement

     8AFB   FO F6         3321            beq  cmd7
                          3322
     8AFD   A9 4E         3323            lda  #$4e
     8AFF   8D 2003       3324            sta  wddat      ; give him the data
     8B02   CA            3325            dex
     8B03   DO EE         3326            bne  cmd7
                          3327
                          3328
     8B05   A0 01         3329   inner    ldy  #1         ; ss
                          3330
     8B07   A2 0C         3331   main7    ldx  #12
     8B09                 3332+           WDTEST          ; chk address
            =0001         3333A           .ife <*!.$03    ; lower two bits cannot be zero
                          3334A           nop             ; fill address error
                          3335A           .endif
     8B09   AD 2000       3336   cmd70    lda  wdstat
     8B0C   29 03         3337            and  #3
     8B0E   4A            3338            lsr  a
     8B0F   90 12         3339            bcc  v1
     8B11   FO F6         3340            beq  cmd70
                          3341
     8B13   A9 00         3342            lda  #0
     8B15   8D 2003       3343            sta  wddat
     8B18   CA            3344            dex
     8B19   DO EE         3345            bne  cmd70
                          3346
     8B1B   A2 03         3347            ldx  #3
     8B1D                 3348+           WDTEST          ; chk address
            =0001         3349A           .ife <*!.$03    ; lower two bits cannot be zero
                          3350A           nop             ; fill address error
                          3351A           .endif
     8B1D   AD 2000       3352   cmd71    lda  wdstat
     8B20   29 03         3353            and  #3
     8B22   4A            3354            lsr  a
     8B23   90 57         3355   v1       bcc  v2
     8B25   FO F6         3356            beq  cmd71
                          3357
     8B27   A9 F5         3358            lda  #$f5
     8B29   8D 2003       3359            sta  wddat
     8B2C   CA            3360            dex
     8B2D   DO EE         3361            bne  cmd71
                          3362
                          3363
     8B2F                 3364+           WDTEST          ; chk address
            =0003         3365A           .ife <*!.$03    ; lower two bits cannot be zero
                          3366A           nop             ; fill address error
                          3367A           .endif
     8B2F   AD 2000       3368   cmd7n    lda  wdstat
     8B32   29 03         3369            and  #3
     8B34   4A            3370            lsr  a
```

```
Error Addr  Code           Seq    Source statement

     8B35   90 45         3371            bcc  v2
     8B37   F0 F6         3372            beq  cmd7n
                         3373
     8B39   A9 FE         3374            lda  #$fe       ; id address mark
     8B3B   8D 2003       3375            sta  wddat
                         3376
                         3377
     8B3E                3378+           WDTEST          ; chk address
            =0002        3379A           .ife <*!.$03    ; lower two bits cannot be zero
                         3380A           nop             ; fill address error
                         3381A           .endif
     8B3E   AD 2000       3382   cmd7f    lda  wdstat
     8B41   29 03         3383            and  #3
     8B43   4A            3384            lsr  a
     8B44   90 36         3385            bcc  v2
     8B46   F0 F6         3386            beq  cmd7f
                         3387
     8B48   AD 01B0       3388            lda  mfmcmd     ; give him the track
     8B4B   8D 2003       3389            sta  wddat
                         3390
     8B4E                3391+           WDTEST          ; chk address
            =0002        3392A           .ife <*!.$03    ; lower two bits cannot be zero
                         3393A           nop             ; fill address error
                         3394A           .endif
     8B4E   AD 2000       3395   cmd7e    lda  wdstat
     8B51   29 03         3396            and  #3
     8B53   4A            3397            lsr  a
     8B54   90 26         3398            bcc  v2
     8B56   F0 F6         3399            beq  cmd7e
                         3400
     8B58   A5 3B         3401            lda  switch
     8B5A   29 10         3402            and  #%00010000 ; what side are we on ?
     8B5C   D0 03         3403            bne  1$
                         3404
     8B5E   A9 00         3405            lda  #0
     8B60   2C            3406            .byte skip2
     8B61   A9 01         3407   1$       lda  #1
     8B63   8D 2003       3408            sta  wddat      ; side number is ...
                         3409
     8B66                3410+           WDTEST          ; chk address
            =0002        3411A           .ife <*!.$03    ; lower two bits cannot be zero
                         3412A           nop             ; fill address error
                         3413A           .endif
     8B66   AD 2000       3414   cmd7d    lda  wdstat
     8B69   29 03         3415            and  #3
     8B6B   4A            3416            lsr  a
     8B6C   90 0E         3417            bcc  v2
     8B6E   F0 F6         3418            beq  cmd7d
                         3419
     8B70   B9 020A       3420            lda  cmdbuf+10,y ; sector number actually cmdbuf+11
```

```
Error Addr  Code           Seq    Source statement

     8B73   8D 2003       3421            sta  wddat
                         3422
     8B76                3423+           WDTEST          ; chk address
            =0002        3424A           .ife <*!.$03    ; lower two bits cannot be zero
                         3425A           nop             ; fill address error
                         3426A           .endif
     8B76   AD 2000       3427   cmd7c    lda  wdstat
     8B79   29 03         3428            and  #3
     8B7B   4A            3429            lsr  a
     8B7C   90 33         3430   v2       bcc  v3
     8B7E   F0 F6         3431            beq  cmd7c
                         3432
     8B80   AD 0205       3433            lda  cmdbuf+5   ; sz
     8B83   8D 2003       3434            sta  wddat
                         3435
     8B86                3436+           WDTEST          ; chk address
            =0002        3437A           .ife <*!.$03    ; lower two bits cannot be zero
                         3438A           nop             ; fill address error
                         3439A           .endif
     8B86   AD 2000       3440   cmd7b    lda  wdstat
     8B89   29 03         3441            and  #3
     8B8B   4A            3442            lsr  a
     8B8C   90 23         3443            bcc  v3
     8B8E   F0 F6         3444            beq  cmd7b
                         3445
     8B90   A9 F7         3446            lda  #$f7       ; crc 2 bytes written
     8B92   8D 2003       3447            sta  wddat
                         3448
     8B95   A2 16         3449            ldx  #22
                         3450
     8B97                3451+           WDTEST          ; chk address
            =0003        3452A           .ife <*!.$03    ; lower two bits cannot be zero
                         3453A           nop             ; fill address error
                         3454A           .endif
     8B97   AD 2000       3455   cmd72    lda  wdstat
     8B9A   29 03         3456            and  #3
     8B9C   4A            3457            lsr  a
     8B9D   90 12         3458            bcc  v3
     8B9F   F0 F6         3459            beq  cmd72
                         3460
                         3461
     8BA1   A9 4E         3462            lda  #$4e
     8BA3   8D 2003       3463            sta  wddat
     8BA6   CA            3464            dex
     8BA7   D0 EE         3465            bne  cmd72
                         3466
     8BA9   A2 0C         3467            ldx  #12
                         3468
     8BAB                3469+           WDTEST          ; chk address
            =0003        3470A           .ife <*!.$03    ; lower two bits cannot be zero
```

```
Error Addr  Code           Seq    Source statement

                          3471A           nop             ; fill address error
                          3472A           .endif
     8BAB   AD 2000       3473   cmd73    lda  wdstat
     8BAE   29 03         3474            and  #3
     8BB0   4A            3475            lsr  a
     8BB1   90 38         3476   v3       bcc  v4
     8BB3   F0 F6         3477            beq  cmd73
                          3478
                          3479
                          3480
     8BB5   A9 00         3481            lda  #0
     8BB7   8D 2003       3482            sta  wddat
     8BBA   CA            3483            dex
     8BBB   D0 EE         3484            bne  cmd73
                          3485
     8BBD   A2 03         3486            ldx  #3
     8BBF                 3487+           WDTEST          ; chk address
            =0003         3488A           .ife <*!.$03    ; lower two bits cannot be zero
                          3489A           nop             ; fill address error
                          3490A           .endif
     8BBF   AD 2000       3491   cmd74    lda  wdstat
     8BC2   29 03         3492            and  #3
     8BC4   4A            3493            lsr  a
     8BC5   90 24         3494            bcc  v4
     8BC7   F0 F6         3495            beq  cmd74
                          3496
                          3497
     8BC9   A9 F5         3498            lda  #$f5       ; a1
     8BCB   8D 2003       3499            sta  wddat
     8BCE   CA            3500            dex
     8BCF   D0 EE         3501            bne  cmd74
                          3502
     8BD1                 3503+           WDTEST          ; chk address
            =0001         3504A           .ife <*1.$03    ; lower two bits cannot be zero
                          3505A           nop             ; fill address error
                          3506A           .endif
     8BD1   AD 2000       3507   cmd7a    lda  wdstat
     8BD4   29 03         3508            and  #3
     8BD6   4A            3509            lsr  a
     8BD7   90 12         3510            bcc  v4
     8BD9   F0 F6         3511            beq  cmd7a
                          3512
     8BDB   A9 FB         3513            lda  #$fb       ; dam
     8BDD   8D 2003       3514            sta  wddat
                          3515
     8BE0   84 6F         3516            sty  temp       ; save current sector
                          3517
                          3518
     8BE2   A4 44         3519            ldy  mfmsiz_hi  ; high
     8BE4                 3520+           WDTEST          ; chk address
```

```
Error Addr  Code           Seq    Source statement

            =0000         3521A           .ife <*!.$03     ; lower two bits cannot be zero
      8BE4  EA            3522A           nop              ; fill address error
                          3523A           .endif
      8BE5  AD 2000       3524    cmd750  lda  wdstat
      8BE8  29 03         3525            and  #3
      8BEA  4A            3526            lsr  a
      8BEB  90 60         3527    v4      bcc  v5
      8BED  F0 F6         3528            beq  cmd750
                          3529
                          3530
      8BEF  AD 020A       3531            lda  cmdbuf+10   ; f1
      8BF2  8D 2003       3532            sta  wddat
                          3533
      8BF5  EC 0271       3534            cpx  mfmsiz_lo
      8BF8  F0 04         3535            beq  cmd75x
                          3536
      8BFA  E8            3537            inx              ; increment
      8BFB  4C 8BE5       3538            jmp  cmd750
                          3539
      8BFE  E8            3540    cmd75x  inx
      8BFF  88            3541            dey
      8C00  D0 E3         3542            bne  cmd750
                          3543
                          3544
      8C02                3545+           WDTEST           ; chk address
            =0002         3546A           .ife <*!.$03     ; lower two bits cannot be zero
                          3547A           nop              ; fill address error
                          3548A           .endif
      8C02  AD 2000       3549    cmd7ff  lda  wdstat
      8C05  29 03         3550            and  #3
      8C07  4A            3551            lsr  a
      8C08  90 43         3552            bcc  v5
      8C0A  F0 F6         3553            beq  cmd7ff
                          3554
      8C0C  A9 F7         3555            lda  #$f7        ; crc
      8C0E  8D 2003       3556            sta  wddat
                          3557
      8C11  AC 0205       3558            ldy  cmdbuf+5    ; ss
      8C14  B9 8C4F       3559            lda  gapmfm,y
      8C17  A4 6F         3560            ldy  temp        ; sector restore
      8C19  AA            3561            tax
                          3562
      8C1A                3563+           WDTEST           ; chk address
            =0002         3564A           .ife <*!.$03     ; lower two bits cannot be zero
                          3565A           nop              ; fill address error
                          3566A           .endif
      8C1A  AD 2000       3567    cmd7fe  lda  wdstat
      8C1D  29 03         3568            and  #3
      8C1F  4A            3569            lsr  a
      8C20  90 2B         3570            bcc  v5
```

```
Error Addr  Code           Seq    Source statement

     8C22   F0 F6         3571            beq  cmd7fe
                         3572
     8C24   A9 4E         3573            lda  #$4e       ; gap 3
     8C26   8D 2003       3574            sta  wddat
     8C29   CA            3575            dex
     8C2A   D0 EE         3576            bne  cmd7fe
                         3577
     8C2C   CC 0207       3578            cpy  cmdbuf+7   ; ns
     8C2F   F0 04         3579            beq  finmfm
                         3580
     8C31   C8            3581            iny             ; inc sector
     8C32   4C 8B07       3582            jmp  main7
                         3583
                         3584
     8C35                 3585+           WDTEST          ; chk address
            =0001         3586A           .ife <*!.$03    ; lower two bits cannot be zero
                         3587A           nop             ; fill address error
                         3588A           .endif
     8C35   AD 2000       3589   finmfm   lda  wdstat
     8C38   29 03         3590            and  #3
     8C3A   4A            3591            lsr  a
     8C3B   90 0B         3592            bcc  vfin
     8C3D   F0 F6         3593            beq  finmfm
                         3594
     8C3F   18            3595            clc
     8C40   A9 4E         3596            lda  #$4e       ; wait for wd to time out
     8C42   8D 2003       3597            sta  wddat
     8C45   4C 8C35       3598            jmp  finmfm
                         3599
     8C48   20 8861       3600   vfin     jsr  waitdn     ; wait for sleepy time
     8C4B   18            3601            clc             ; good carry
     8C4C   24            3602            .byte skip1
     8C4D   38            3603   v5       sec
     8C4E   60            3604            rts
                         3605
                         3606            ; min gap size @2% fast
     8C4F   07 0C 17      3607   gapmfm   .byte 7, 12, 23, 44
     8C52   2C
                         3608
     8C53   1A 10 09      3609   numsek   .byte 26, 16, 9, 5
     8C56   05
                         3610
```

```
Error Addr  Code           Seq    Source statement

                           3612   ; format disk
                           3613
                           3614   ;  0     1    2    3      4    5    6    7    8   9   10  + cmdbuf
                           3615   ;  ^     ^    ^    ^      ^    ^    ^    ^    ^   ^   ^
                           3616   ; "U" + $00 + N +  MD  + CP + SZ + LT + NS + ST + S + FL
                           3617
     8C57   A5 3B          3618   cmdeig  lda  switch     ; check abort command switch
     8C59   29 08          3619           and  #%00001000
     8C5B   F0 07          3620           beq  1$
                           3621
     8C5D   A6 46          3622           ldx  ctl_dat    ; get error
     8C5F   8E 01B0        3623           stx  mfmcmd     ; save for prcmd
     8C62   38             3624           sec             ; wp error usually
     8C63   60             3625           rts
                           3626
     8C64   20 D307        3627   1$      jsr  clrchn     ; close all channels
     8C67   AD 0274        3628           lda  cmdsiz     ; setup default parms
     8C6A   38             3629           sec
     8C6B   E9 04          3630           sbc  #4         ; less mandatory + 1
     8C6D   A8             3631           tay
     8C6E   F0 20          3632           beq  cp00       ; mode only, gave cp
                           3633
     8C70   88             3634           dey
     8C71   F0 22          3635           beq  sz00       ; md, cp only, gave sz
                           3636
     8C73   A9 00          3637           lda  #0
     8C75   8D 01B0        3638           sta  mfmcmd     ; clear status
                           3639
     8C78   AD 0205        3640           lda  cmdbuf+5
     8C7B   20 8A57        3641           jsr  consek     ; setup sector size
                           3642
     8C7E   88             3643           dey
     8C7F   F0 21          3644           beq  lt00       ; md, cp, sz only, gave lt
                           3645
     8C81   88             3646           dey
     8C82   F0 23          3647           beq  ns00       ; md, cp, sz, lt only, gave ns
                           3648
     8C84   88             3649           dey
     8C85   F0 26          3650           beq  st00       ; md, cp, sz, lt, ns only, gave st
                           3651
     8C87   88             3652           dey
     8C88   F0 2B          3653           beq  s00        ; md, cp, sz, lt, ns, st only, gave s
                           3654
     8C8A   88             3655           dey
     8C8B   F0 2D          3656           beq  fl00       ; md, cp, sz, lt, ns, st, ss only, gave fl
                           3657
     8C8D   4C 8CBF        3658           jmp  start8
                           3659
```

```
Error Addr  Code          Seq    Source statement

     8C90   A9 00        3661   cp00    lda  #0          ; default interleave
     8C92   8D 0204      3662           sta  cmdbuf+4
                         3663
     8C95   A9 00        3664   sz00    lda  #0
     8C97   8D 01B0      3665           sta  mfmcmd      ; clear status
                         3666
     8C9A   A9 01        3667           lda  #1          ; 256 byte sectors
     8C9C   8D 0205      3668           sta  cmdbuf+5
     8C9F   20 8A57      3669           jsr  consek      ; setup block size
                         3670
     8CA2   A9 27        3671   lt00    lda  #39         ; last track is #39, 40 tracks total
     8CA4   8D 0206      3672           sta  cmdbuf+6
                         3673
     8CA7   BD 8C53      3674   ns00    lda  numsek,x    ; x=sector size index for # of sectors per track
     8CAA   8D 0207      3675           sta  cmdbuf+7
                         3676
     8CAD   A9 00        3677   st00    lda  #0          ; default track #0 start
     8CAF   8D 0208      3678           sta  cmdbuf+8
     8CB2   8D 2001      3679           sta  wdtrk
                         3680
     8CB5   A9 00        3681   s00     lda  #0          ; default steps from track 00
     8CB7   8D 0209      3682           sta  cmdbuf+9
                         3683
     8CBA   A9 E5        3684   fl00    lda  #$e5        ; default block fill
     8CBC   8D 020A      3685           sta  cmdbuf+10
                         3686
```

```
Error Addr  Code           Seq    Source statement

     8CBF   20 8CDE       3688   start8  jsr  go_fmt      ; format side zero
     8CC2   AD 01B0       3689           lda  mfmcmd      ; error ?
     8CC5   E0 02         3690           cpx  #2
     8CC7   B0 12         3691           bcs  1$
                          3692
     8CC9   A5 3B         3693           lda  switch      ; check for double sided opt
     8CCB   29 20         3694           and  #%00100000
     8CCD   F0 0C         3695           beq  1$
                          3696
     8CCF   A5 3B         3697           lda  switch      ; set single side
     8CD1   09 10         3698           ora  #%00010000
     8CD3   85 3B         3699           sta  switch
     8CD5   20 8954       3700           jsr  sel_sid     ; select h/w
     8CD8   20 8CDE       3701           jsr  go_fmt      ; format side one
     8CDB   4C 89EF       3702   1$      jmp  cmdone      ; restore
                          3703
```

```
Error Addr  Code           Seq    Source statement

     8CDE   20 89B3       3705   go_fmt  jsr  diskin     ; is there a diskette in the unit
     8CE1   B0 7C         3706          bcs  c_801      ; br, nope...
                          3707
     8CE3   A9 01         3708          lda  #1
     8CE5   8D 180D       3709          sta  ifr1       ; clear irq from write protect
     8CE8   20 89EF       3710          jsr  cmdone     ; restore to track one/zero ?
     8CEB   AD 0208       3711          lda  cmdbuf+8   ; store logical
     8CEE   8D 01B0       3712          sta  mfmcmd     ; in mfmcmd
     8CF1   8D 2001       3713          sta  wdtrk
                          3714
     8CF4   2C 0203       3715          bit  cmdbuf+3   ; check table bit
     8CF7   70 05         3716          bvs  1$         ; br, table has been given to us
                          3717
     8CF9   20 8888       3718          jsr  sectcv     ; generate sector table
     8CFC   B0 61         3719          bcs  c_801
                          3720
     8CFE   AD 0209       3721   1$     lda  cmdbuf+9   ; offset track 00 by s steps
     8D01   29 7F         3722          and  #$7f       ; clear mode bit
     8D03   F0 08         3723          beq  c_800
                          3724
     8D05   18            3725          clc
     8D06   65 67         3726          adc  cmd_trk    ; add to it
     8D08   85 67         3727          sta  cmd_trk    ; physical
     8D0A   20 87BA       3728          jsr  seke       ; & seek to it
                          3729
     8D0D   78            3730   c_800  sei
     8D0E   AD 180D       3731          lda  ifr1       ; check for disk change
     8D11   4A            3732          lsr  a
     8D12   B0 4B         3733          bcs  c_801
                          3734
     8D14   20 8A86       3735          jsr  cmdsev     ; format track
     8D17   B0 46         3736          bcs  c_801
                          3737
     8D19   AD 180D       3738          lda  ifr1       ; check for disk change
     8D1C   4A            3739          lsr  a
     8D1D   B0 40         3740          bcs  c_801
                          3741
     8D1F   20 88F0       3742          jsr  verfmt     ; verify format
     8D22   B0 3B         3743          bcs  c_801
                          3744
     8D24   AD 180D       3745          lda  ifr1       ; check for disk change
     8D27   4A            3746          lsr  a
     8D28   B0 35         3747          bcs  c_801
                          3748
     8D2A   AD 01B0       3749          lda  mfmcmd     ; last track ?
     8D2D   CD 0206       3750          cmp  cmdbuf+6   ; lt
     8D30   F0 0E         3751          beq  c_802
                          3752
     8D32   E6 67         3753          inc  cmd_trk
     8D34   EE 2001       3754          inc  wdtrk
```

```
Error Addr  Code          Seq    Source statement

     8D37   EE 01B0      3755            inc  mfmcmd      ; track to write
                         3756
     8D3A   20 87BA      3757            jsr  seke        ; goto next track
     8D3D   4C 8D0D      3758            jmp  c_800
                         3759
     8D40   24 3B        3760    c_802   bit  switch
     8D42   10 18        3761            bpl c_803        ; kill next
                         3762
     8D44   38           3763            sec
     8D45   AD 0206      3764            lda  cmdbuf+6    ; lt
     8D48   ED 0208      3765            sbc  cmdbuf+8    ; st
     8D4B   C9 27        3766            cmp  #39         ; no more than 40 tracks
     8D4D   B0 0D        3767            bcs  c_803
                         3768
     8D4F   E6 67        3769            inc  cmd_trk     ; clear next track
     8D51   20 87BA      3770            jsr  seke
     8D54   A2 1C        3771            ldx  #28
     8D56   20 9D63      3772            jsr  jclear      ; with gcr 0
     8D59   20 FE00      3773            jsr  kill        ; read mode again
                         3774
     8D5C   A2 00        3775    c_803   ldx  #0
     8D5E   2C           3776            .byte  skip2
     8D5F   A2 06        3777    c_801   ldx  #6
     8D61   8E 01B0      3778            stx  mfmcmd      ; ok exit
     8D64   4C 85E9      3779            jmp  upinst
                         3780            .include  mfmsubr2
```

```
Error Addr  Code           Seq    Source statement

                          3782            .subttl 'mfmsubr2.src'
                          3783
                          3784    ;  mfm read
                          3785
     8D67   A5 3B         3786    cmdnin  lda  switch
     8D69   29 20         3787            and  #%00100000
     8D6B   D0 59         3788            bne  c_904
                          3789
     8D6D   A9 03         3790            lda  #3
     8D6F   85 31         3791            sta  bufpnt+1    ; buffer #1
     8D71   A0 00         3792            ldy  #0          ; even page
     8D73   84 30         3793            sty  bufpnt
     8D75   A6 44         3794            ldx  mfmsiz_hi   ; get sector size
     8D77   AD 0203       3795            lda  cmdbuf+3    ; get track address
     8D7A   8D 2001       3796            sta  wdtrk       ; give track to wd
     8D7D   AD 0204       3797            lda  cmdbuf+4    ; get sector address
     8D80   8D 2002       3798            sta  wdsec       ; give sector to wd
     8D83   A9 88         3799            lda  #$88        ; read sector command
     8D85   20 884E       3800            jsr  strtwd      ; send cmd
                          3801
     8D88                 3802+           WDTEST           ; chk address
            =0000         3803A           .ife <*!.$03     ; lower two bits cannot be zero
     8D88   EA            3804A           nop              ; fill address error
                          3805A           .endif
     8D89   AD 2000       3806    c_900   lda  wdstat      ; get a byte
     8D8C   29 03         3807            and  #3
     8D8E   4A            3808            lsr  a
     8D8F   90 1A         3809            bcc  c_902
                          3810
     8D91   29 01         3811            and  #1
     8D93   F0 F4         3812            beq  c_900
                          3813
     8D95   AD 2003       3814            lda  wddat       ; get a byte from wd
     8D98   91 30         3815            sta  (bufpnt),y  ; put it in the buffer
     8D9A   CC 0271       3816            cpy  mfmsiz_lo   ; done buffer
     8D9D   F0 03         3817            beq  c_901       ; next
                          3818
     8D9F   C8            3819            iny
     8DA0   D0 E7         3820            bne  c_900
                          3821
     8DA2   C8            3822    c_901   iny              ; y=0
     8DA3   CA            3823            dex
     8DA4   F0 05         3824            beq  c_902
                          3825
     8DA6   E6 31         3826            inc  bufpnt+1    ; next buffer
     8DA8   4C 8D89       3827            jmp  c_900
                          3828
     8DAB   20 8861       3829    c_902   jsr  waitdn      ; wait for unbusy
                          3830
     8DAE   20 883C       3831            jsr  cvstat      ; convert
```

```
Error Addr  Code           Seq    Source statement

     8DB1   20 85E9       3832            jsr  upinst      ; update controller status
                          3833
     8DB4   24 3B         3834            bit  switch
     8DB6   70 07         3835            bvs  c_905       ; ignore error ?
                          3836
     8DB8   E0 02         3837            cpx  #2
     8DBA   90 03         3838            bcc  c_905
                          3839
     8DBC   4C 8384       3840            jmp  finbad      ; send it
                          3841
     8DBF   20 85F9       3842  c_905     jsr  hskrd       ; send status
                          3843
     8DC2   A5 3B         3844            lda  switch
     8DC4   30 22         3845            bmi  c_906       ; buffer transfer ?
                          3846
     8DC6   A9 03         3847  c_904     lda  #3
     8DC8   85 31         3848            sta  bufpnt+1    ; buffer #1
     8DCA   A0 00         3849            ldy  #0          ; even page
     8DCC   84 30         3850            sty  bufpnt
     8DCE   A6 44         3851            ldx  mfmsiz_hi   ; get sector size
     8DD0   B1 30         3852  c_907     lda  (bufpnt),y  ; get data from buffer
     8DD2   85 46         3853            sta  ctl_dat
     8DD4   20 85F9       3854            jsr  hskrd       ; handshake data to host
                          3855
     8DD7   CC 0271       3856            cpy  mfmsiz_lo   ; done buffer ?
     8DDA   F0 03         3857            beq  c_908       ; next
                          3858
     8DDC   C8            3859            iny
     8DDD   D0 F1         3860            bne  c_907
                          3861
     8DDF   C8            3862  c_908     iny              ; y=0
     8DE0   CA            3863            dex
     8DE1   F0 05         3864            beq  c_906
                          3865
     8DE3   E6 31         3866            inc  bufpnt+1    ; next buffer
     8DE5   4C 8DD0       3867            jmp  c_907
                          3868
     8DE8   CE 0205       3869  c_906     dec  cmdbuf+5    ; any more sectors ?
     8DEB   F0 06         3870            beq  cmd9ex
                          3871
     8DED   20 886C       3872            jsr  sectnx      ; next sector rap ?
     8DF0   4C 8D67       3873            jmp  cmdnin      ; continue...
                          3874
     8DF3   4C 891B       3875  cmd9ex    jmp  chcsee      ; next track
                          3876
```

```
Error Addr  Code           Seq    Source statement

                          3878   ; mfm write
                          3879
     8DF6   A9 03         3880   cmdten  lda  #3
     8DF8   85 31         3881           sta  bufpnt+1   ; buffer #1
     8DFA   A0 00         3882           ldy  #0
     8DFC   84 30         3883           sty  bufpnt
     8DFE   A6 44         3884           ldx  mfmsiz_hi  ; get high byte
     8E00   A5 3B         3885           lda  switch
     8E02   30 30         3886           bmi  c_100      ; buffer transfer ?
                          3887
     8E04   AD 1800       3888   c_101   lda  pb          ; debounce
     8E07   49 08         3889           eor  #clkout    ; toggle clock
     8E09   2C 400D       3890           bit  icr        ; clear pending
     8E0C   8D 1800       3891           sta  pb         ; doit
                          3892
     8E0F   AD 1800       3893   c_103   lda  pb
     8E12   10 03         3894           bpl  c_102      ; atn ?
                          3895
     8E14   20 EA59       3896           jsr  tstatn     ; service if appropiate
                          3897
     8E17   AD 400D       3898   c_102   lda  icr
     8E1A   29 08         3899           and  #8
     8E1C   F0 F1         3900           beq  c_103      ; wait for byte ready
                          3901
     8E1E   AD 400C       3902           lda  sdr        ; get data
     8E21   91 30         3903           sta  (bufpnt),y
     8E23   CC 0271       3904           cpy  mfmsiz_lo
     8E26   F0 03         3905           beq  c_104
                          3906
     8E28   C8            3907           iny
     8E29   D0 D9         3908           bne  c_101
                          3909
     8E2B   C8            3910   c_104   iny             ; y=0
     8E2C   CA            3911           dex
     8E2D   F0 05         3912           beq  c_100
                          3913
     8E2F   E6 31         3914           inc  bufpnt+1   ; next buffer
     8E31   4C 8E04       3915           jmp  c_101
                          3916
     8E34   A5 3B         3917   c_100   lda  switch
     8E36   29 20         3918           and  #%00100000
     8E38   D0 7D         3919           bne  c_105      ; buffer op
                          3920
     8E3A   A5 3B         3921           lda  switch
     8E3C   29 08         3922           and  #%00001000 ; check internal switch
     8E3E   F0 05         3923           beq  c_112
                          3924
     8E40   A6 46         3925           ldx  ctl_dat
     8E42   4C 8381       3926           jmp  fail
                          3927
```

```
Error Addr  Code           Seq    Source statement

     8E45   A9 03         3928   c_112   lda  #3          ; write now
     8E47   85 31         3929           sta  bufpnt+1    ; buffer #1
     8E49   A0 00         3930           ldy  #0
     8E4B   84 30         3931           sty  bufpnt
     8E4D   A6 44         3932           ldx  mfmsiz_hi
     8E4F   AD 0203       3933           lda  cmdbuf+3    ; get track
     8E52   8D 2001       3934           sta  wdtrk
     8E55   AD 0204       3935           lda  cmdbuf+4
     8E58   8D 2002       3936           sta  wdsec
     8E5B   AD 180D       3937           lda  ifr1        ; check for diskette change
     8E5E   4A            3938           lsr  a
     8E5F   B0 32         3939           bcs  c_110
                          3940
     8E61   A9 A8         3941           lda  #$a8        ; normal dam
     8E63   20 884E       3942           jsr  strtwd
                          3943
     8E66                 3944+          WDTEST           ; chk address
            =0002         3945A          .ife <*!.$03     ; lower two bits cannot be zero
                          3946A          nop              ; fill address error
                          3947A          .endif
     8E66   AD 2000       3948   cmd10   lda  wdstat
     8E69   29 03         3949           and  #3
     8E6B   4A            3950           lsr  a
     8E6C   90 25         3951           bcc  c_110
                          3952
     8E6E   29 01         3953           and  #1
     8E70   F0 F4         3954           beq  cmd10
                          3955
     8E72   B1 30         3956           lda  (bufpnt),y
     8E74   8D 2003       3957           sta  wddat       ; send data to wd1770
     8E77   CC 0271       3958           cpy  mfmsiz_lo
     8E7A   F0 03         3959           beq  cmd100
                          3960
     8E7C   C8            3961           iny
     8E7D   D0 E7         3962           bne  cmd10
                          3963
     8E7F   C8            3964   cmd100  iny              ; y=0
     8E80   CA            3965           dex
     8E81   F0 05         3966           beq  c_108
                          3967
     8E83   E6 31         3968           inc  bufpnt+1    ; next page...
     8E85   4C 8E66       3969           jmp  cmd10
                          3970
     8E88   AD 180D       3971   c_108   lda  ifr1        ; check for diskette change
     8E8B   4A            3972           lsr  a
     8E8C   B0 05         3973           bcs  c_110
                          3974
     8E8E   20 8EC6       3975           jsr  cmdele      ; verify
     8E91   90 07         3976           bcc  c_109
                          3977
```

```
Error Addr  Code          Seq   Source statement

     8E93   20 81CE      3978   c_110   jsr  spout       ; go output
     8E96   A2 07        3979           ldx  #7
     8E98   D0 06        3980           bne  c_111       ; bra
                         3981
     8E9A   20 81CE      3982   c_109   jsr  spout
     8E9D   20 883C      3983           jsr  cvstat
     8EA0   8E 01B0      3984   c_111   stx  mfmcmd      ; save status
     8EA3   20 85E9      3985           jsr  upinst
     8EA6   20 85F9      3986           jsr  hskrd       ; send it
     8EA9   20 86A0      3987           jsr  burst       ; wait for clk high
     8EAC   20 81B2      3988           jsr  spinp       ; input
                         3989
     8EAF   24 3B        3990           bit  switch      ; abort on error ?
     8EB1   70 04        3991           bvs  c_105
                         3992
     8EB3   E0 02        3993           cpx  #2          ; error ?
     8EB5   B0 0E        3994           bcs  c_107
                         3995
     8EB7   CE 0205      3996   c_105   dec  cmdbuf+5    ; more sectors
     8EBA   F0 06        3997           beq  c_106
                         3998
     8EBC   20 886C      3999           jsr  sectnx      ; next sector
     8EBF   4C 8DF6      4000           jmp  cmdten
                         4001
     8EC2   4C 891B      4002   c_106   jmp  chcsee
                         4003
     8EC5   60           4004   c_107   rts
                         4005           .include  mfmsubr3
```

```
Error Addr  Code           Seq    Source statement

                          4007            .subttl 'mfmsubr3.src'
                          4008
                          4009    ; mfm verify for write operation
                          4010
     8EC6   A9 03         4011    cmdele  lda  #3
     8EC8   85 31         4012            sta  bufpnt+1   ; buffer #0
     8ECA   A0 00         4013            ldy  #0         ; even page
     8ECC   84 30         4014            sty  bufpnt
     8ECE   A6 44         4015            ldx  mfmsiz_hi  ; get sector high
     8ED0   AD 0203       4016            lda  cmdbuf+3   ; get track address
     8ED3   8D 2001       4017            sta  wdtrk      ; give track to wd
     8ED6   AD 0204       4018            lda  cmdbuf+4   ; get sector address
     8ED9   8D 2002       4019            sta  wdsec      ; give sector to wd
     8EDC   A9 88         4020    cmd_11  lda  #$88       ; read sector command
     8EDE   20 884E       4021            jsr  strtwd     ; send cmd
                          4022
     8EE1                 4023+           WDTEST          ; chk address
            =0001         4024A           .ife <*!.$03    ; lower two bits cannot be zero
                          4025A           nop             ; fill address error
                          4026A           .endif
     8EE1   AD 2000       4027    c_1100  lda  wdstat     ; get a byte
     8EE4   29 03         4028            and  #3
     8EE6   4A            4029            lsr  a
     8EE7   90 1C         4030            bcc  c_1101
                          4031
     8EE9   29 01         4032            and  #1
     8EEB   F0 F4         4033            beq  c_1100     ; byte ready
                          4034
     8EED   AD 2003       4035            lda  wddat      ; get a byte from wd
     8EF0   D1 30         4036            cmp  (bufpnt),y ; put it in the buffer
     8EF2   D0 11         4037            bne  c_1101     ; flag error
                          4038
     8EF4   CC 0271       4039            cpy  mfmsiz_lo  ; done buffer
     8EF7   F0 03         4040            beq  c_1102     ; next
                          4041
     8EF9   C8            4042            iny
     8EFA   D0 E5         4043            bne  c_1100
                          4044
     8EFC   C8            4045    c_1102  iny             ; y=0
     8EFD   CA            4046            dex
     8EFE   F0 10         4047    c_1103  beq  c_1104
                          4048
     8F00   E6 31         4049            inc  bufpnt+1   ; next buffer
     8F02   4C 8EE1       4050            jmp  c_1100
                          4051
     8F05   A9 D0         4052    c_1101  lda  #$d0       ; abort wd1770
     8F07                 4053+           WDTEST          ; chk address
            =0003         4054A           .ife <*!.$03    ; lower two bits cannot be zero
                          4055A           nop             ; fill address error
                          4056A           .endif
```

```
Error Addr  Code           Seq    Source statement

     8F07   8D 2000       4057            sta  wdstat     ; send cmd
                          4058
     8F0A   20 A483       4059            jsr  jslowd     ; wait 40 uS.
                          4060
     8F0D   A2 07         4061            ldx  #7
     8F0F   2C            4062            .byte skip2
     8F10   A2 00         4063   c_1104   ldx  #0
     8F12   8E 01B0       4064            stx  mfmcmd
     8F15   4C 8861       4065            jmp  waitdn     ; wait for the wd1770 to sleep
                          4066
```

```
Error Addr  Code           Seq    Source statement

                           4068   ; verify sector to fill byte
                           4069   ; track and sector must be setup previous
                           4070
     8F18   A9 03          4071   cmdtwv  lda  #3
     8F1A   85 31          4072           sta  bufpnt+1    ; buffer #1
     8F1C   A0 00          4073           ldy  #0          ; even page
     8F1E   84 30          4074           sty  bufpnt
     8F20   A6 44          4075           ldx  mfmsiz_hi   ; get sector high
     8F22   AC 0271        4076           ldy  mfmsiz_lo   ; get lo sector size
     8F25   A9 88          4077   cmd_12  lda  #$88        ; read sector command
     8F27   20 884E        4078           jsr  strtwd      ; send cmd
                           4079
     8F2A                  4080+          WDTEST           ; chk address
            =0002          4081A          .ife <*!.$03     ; lower two bits cannot be zero
                           4082A          nop              ; fill address error
                           4083A          .endif
     8F2A   AD 2000        4084   c_1200  lda  wdstat      ; get a byte
     8F2D   29 03          4085           and  #3
     8F2F   4A             4086           lsr  a
     8F30   90 1A          4087           bcc  c_1201
                           4088
     8F32   29 01          4089           and  #1
     8F34   F0 F4          4090           beq  c_1200      ; byte ready
                           4091
     8F36   AD 2003        4092           lda  wddat       ; get a byte from wd
     8F39   CD 020A        4093           cmp  cmdbuf+10   ; same as fill byte ?
     8F3C   D0 0E          4094           bne  c_1201      ; br, on error
                           4095
     8F3E   88             4096           dey
     8F3F   10 E9          4097           bpl  c_1200      ; done ?
                           4098
     8F41   CA             4099   c_1202  dex
     8F42   F0 13          4100   c_1203  beq  c_1204
                           4101
     8F44   AC 0271        4102           ldy  mfmsiz_lo   ; get it again
     8F47   E6 31          4103           inc  bufpnt+1    ; next buffer
     8F49   4C 8F2A        4104           jmp  c_1200
                           4105
     8F4C   A9 D0          4106   c_1201  lda  #$d0        ; abort wd1770
     8F4E                  4107+          WDTEST           ; chk address
            =0002          4108A          .ife <*!.$03     ; lower two bits cannot be zero
                           4109A          nop              ; fill address error
                           4110A          .endif
     8F4E   8D 2000        4111           sta  wdstat      ; send cmd
                           4112
     8F51   20 A483        4113           jsr  jslowd      ; wait 40 uS.
                           4114
     8F54   A2 07          4115           ldx  #7
     8F56   2C             4116           .byte skip2
     8F57   A2 00          4117   c_1204  ldx  #0
```

```
Error Addr  Code           Seq    Source statement

     8F59   8E 01B0       4118            stx  mfmcmd
     8F5C   4C 8861       4119            jmp  waitdn      ; wait for the wd1770 to sleep
                          4120
```

```
Error Addr  Code           Seq    Source statement

     8F5F   08             4122   cmdthi  php
     8F60   78             4123           sei
     8F61   20 89EF        4124           jsr  cmdone      ; track zero start
                           4125
     8F64   24 3B          4126           bit  switch     ; seek to n-track ?
     8F66   10 08          4127           bpl  1$
                           4128
     8F68   AD 0203        4129           lda  cmdbuf+3
     8F6B   85 67          4130           sta  cmd_trk    ; goto this track
     8F6D   20 87BA        4131           jsr  seke
                           4132
     8F70   A9 00          4133   1$      lda  #0
     8F72   85 97          4134           sta  cpmsek     ; clear # of sectors
                           4135
     8F74   20 8A27        4136           jsr  cmdfiv     ; read address
     8F77   AE 01B0        4137           ldx  mfmcmd     ; check status
     8F7A   E0 02          4138           cpx  #2
     8F7C   B0 1F          4139           bcs  2$
                           4140
     8F7E   A5 26          4141           lda  mfmhdr+2
     8F80   85 96          4142           sta  cmd_sec    ; this is where we stop
                           4143
     8F82   20 8A27        4144   3$      jsr  cmdfiv     ; read address
     8F85   A5 26          4145           lda  mfmhdr+2   ; get next sector
     8F87   A4 97          4146           ldy  cpmsek
     8F89   99 020B        4147           sta  cmdbuf+11,y
     8F8C   E6 97          4148           inc  cpmsek     ; inc sector count
     8F8E   C0 1F          4149           cpy  #31        ; went too far ?
     8F90   B0 0B          4150           bcs  2$
                           4151
     8F92   C5 96          4152           cmp  cmd_sec    ; done yet ?
     8F94   D0 EC          4153           bne  3$         ; wait for rap...
                           4154
     8F96   A5 24          4155           lda  mfmhdr     ; get track
     8F98   85 67          4156           sta  cmd_trk    ; save for later
                           4157
     8F9A   A2 00          4158           ldx #0
     8F9C   2C             4159           .byte skip2
     8F9D   A2 02          4160   2$      ldx #2
     8F9F   8E 01B0        4161           stx  mfmcmd
     8FA2   28             4162           plp
     8FA3   60             4163           rts
                           4164           .include  utload
```

```
Error Addr  Code           Seq    Source statement

                          4166            .subttl 'utload.src'
                          4167
                          4168    ;***************************
                          4169    ;**                       **
                          4170    ;**     1571  UTILITY     **
                          4171    ;**                       **
                          4172    ;** * * * * * * * * * * * **
                          4173    ;** UO n M  = mode        **
                          4174    ;** UO n S  = int. dos     **
                          4175    ;** UO n R  = retries     **
                          4176    ;** UO n H  = head 0,1    **
                          4177    ;** UO n T  = signature   **
                          4178    ;** UO n #  = device      **
                          4179    ;**                       **
                          4180    ;**     n = ">" ascii     **
                          4181    ;**                       **
                          4182    ;***************************
                          4183
     8FA4   AD 0204       4184    cmdsec  lda  cmdbuf+4
     8FA7   85 69         4185            sta  secinc
     8FA9   60            4186            rts
                          4187
     8FAA   AD 0204       4188    cmdret  lda  cmdbuf+4
     8FAD   85 6A         4189            sta  revcnt
     8FAF   60            4190            rts
                          4191
                          4192
     8FB0   4C 924E       4193    sign    jmp  signature  ; finish up there
                          4194
                          4195
     8FB3   78            4196    heads   sei             ; no irq's
     8FB4   AD 180F       4197            lda  pota1
     8FB7   29 20         4198            and  #$20
     8FB9   D0 66         4199            bne  utlbad     ; 1541 only
                          4200
     8FBB   AD 0204       4201            lda  cmdbuf+4   ; which ?
     8FBE   C9 31         4202            cmp  #'1
     8FC0   F0 12         4203            beq  1$
                          4204
     8FC2   C9 30         4205            cmp  #'0
     8FC4   D0 5B         4206            bne  utlbad     ; no drive # specified
                          4207
     8FC6   AD 180F       4208            lda  pota1      ; 1541 mode
     8FC9   29 FB         4209            and  #$ff-$04   ; side 0
     8FCB   8D 180F       4210            sta  pota1
     8FCE   58            4211            cli
     8FCF   24 3B         4212            bit  switch
     8FD1   10 0E         4213            bpl  2$
                          4214
     8FD3   60            4215            rts
```

```
Error Addr  Code           Seq    Source statement

                          4216
      8FD4   AD 180F      4217   1$       lda  pota1
      8FD7   09 04        4218            ora  #$04
      8FD9   8D 180F      4219            sta  pota1      ; side 1
      8FDC   58           4220            cli
      8FDD   24 3B        4221            bit  switch
      8FDF   30 03        4222            bmi  3$
                          4223
      8FE1   4C D042      4224   2$       jmp  initdr
                          4225
      8FE4   60           4226   3$       rts
                          4227
      8FE5   AE 0274      4228   chgutl   ldx  cmdsiz     ; chk cmd size
      8FE8   E0 04        4229            cpx  #4
      8FEA   90 35        4230            bcc  utlbad     ; br, error no parameters
                          4231
      8FEC   AD 0203      4232            lda  cmdbuf+3
      8FEF   C9 53        4233            cmp  #'S        ; sector interleave ?
      8FF1   F0 B1        4234            beq  cmdsec
                          4235
      8FF3   C9 52        4236            cmp  #'R        ; retry
      8FF5   F0 B3        4237            beq  cmdret
                          4238
      8FF7   C9 54        4239            cmp  #'T        ; test ROM
      8FF9   F0 B5        4240            beq  sign
                          4241
      8FFB   C9 4D        4242            cmp  #'M        ; 1571/1541 Mode
      8FFD   F0 27        4243            beq  modes
                          4244
      8FFF   C9 48        4245            cmp  #'H        ; 1541 head change
      9001   F0 B0        4246            beq  heads
                          4247
      9003   A8           4248            tay             ; save
      9004   C0 04        4249            cpy  #4         ; between 4-30
      9006   90 19        4250            bcc  utlbad
                          4251
      9008   C0 1F        4252            cpy  #31
      900A   B0 15        4253            bcs  utlbad
                          4254
      900C   A9 40        4255            lda  #$40       ; change device #
      900E   85 78        4256            sta  tlkadr     ; clear old
                          4257
      9010   A9 20        4258            lda  #$20
      9012   85 77        4259            sta  lsnadr     ; *
                          4260
      9014   98           4261            tya
      9015   18           4262            clc
      9016   65 78        4263            adc  tlkadr
      9018   85 78        4264            sta  tlkadr     ; new
      901A   98           4265            tya
```

```
Error Addr  Code           Seq    Source statement

     901B   18             4266           clc
     901C   65 77          4267           adc  lsnadr
     901E   85 77          4268           sta  lsnadr      ; new
     9020   60             4269           rts
                           4270
     9021   A9 31          4271   utlbad  lda  #badcmd
     9023   4C C1C8        4272           jmp  cmderr
                           4273
     9026   78             4274   modes   sei              ; no irq's
     9027   AD 0204        4275           lda  cmdbuf+4    ; which ?
     902A   C9 31          4276           cmp  #'1
     902C   F0 20          4277           beq  1$
                           4278
     902E   C9 30          4279           cmp  #'0
     9030   D0 EF          4280           bne  utlbad      ; br, bad
                           4281
     9032   AD 180F        4282           lda  pota1       ; 1541 mode
     9035   29 DF          4283           and  #$ff-$20    ; 1Mhz
     9037   8D 180F        4284           sta  pota1
     903A   20 A483        4285           jsr  jslowd      ; wait a while
     903D   20 FF82        4286           jsr  ptch10      ; init
     9040   AD 02AF        4287           lda  lock
     9043   09 80          4288           ora  #$80
     9045   8D 02AF        4289           sta  lock        ; auto off
     9048   58             4290           cli
     9049   24 3B          4291           bit  switch
     904B   10 2F          4292           bpl  2$
                           4293
     904D   60             4294           rts
                           4295
     904E   AD 180F        4296   1$      lda  pota1       ; 1571 mode
     9051   09 20          4297           ora  #$20
     9053   8D 180F        4298           sta  pota1       ; 2 Mhz
     9056   20 A483        4299           jsr  jslowd      ; wait a bit
     9059   A9 DE          4300           lda  #<jirq
     905B   8D 02A9        4301           sta  irqjmp      ; revector irq
     905E   A9 9D          4302           lda  #>jirq
     9060   8D 02AA        4303           sta  irqjmp+1
     9063   A9 40          4304           lda  #tim2       ; setup timers
     9065   8D 1C07        4305           sta  t1hl2
     9068   8D 1C05        4306           sta  t1hc2       ; for 2 Mhz
     906B   AD 02AF        4307           lda  lock
     906E   29 7F          4308           and  #$ff-$80
     9070   8D 02AF        4309           sta  lock        ; auto on
     9073   A9 00          4310           lda  #0
     9075   85 62          4311           sta  nxtst       ; used as a pointer in 1541 mode
     9077   58             4312           cli
     9078   24 3B          4313           bit  switch
     907A   30 03          4314           bmi  3$
                           4315
```

```
Error Addr  Code           Seq    Source statement

     907C   4C D042       4316   2$       jmp  initdr
                          4317
     907F   60            4318   3$       rts
                          4319            .include  fastld
```

```
Error Addr  Code          Seq   Source statement

                          4321          .subttl 'fastld.src'
                          4322
     9080   20 81CE       4323  fstload jsr  spout      ; output
     9083   20 91EA       4324          jsr  set_fil    ; setup filename for parser
     9086   B0 5F         4325          bcs  9$
                          4326
     9088   20 C63D       4327          jsr  autoi      ; init mechanism
     908B   A5 FF         4328          lda  nodrv      ; chk status
     908D   D0 58         4329          bne  9$         ; no drive status
                          4330
     908F   A5 37         4331          lda  fastsr     ; set error recovery flag on
     9091   09 81         4332          ora  #$81       ; & eoi flag
     9093   85 37         4333          sta  fastsr
     9095   20 91CA       4334          jsr  findbuf    ; check for buffer availabilty
                          4335
     9098   AD 0200       4336          lda  cmdbuf
     909B   C9 2A         4337          cmp  #'*        ; load last ?
     909D   D0 0F         4338          bne  7$
                          4339
     909F   A5 7E         4340          lda  prgtrk     ; any file ?
     90A1   F0 0B         4341          beq  7$
                          4342
     90A3   48            4343          pha             ; save track
     90A4   AD 026F       4344          lda  prgsec
     90A7   8D 0285       4345          sta  filsec     ; update
     90AA   68            4346          pla
     90AB   4C 90EC       4347          jmp  1$
                          4348
     90AE   A9 00         4349  7$      lda  #0
     90B0   A8            4350          tay
     90B1   AA            4351          tax             ; clear .a, .x, .y
     90B2   8D 028E       4352          sta  lstdrv     ; init drive number
     90B5   8D 027A       4353          sta  filtbl     ; set up for file name parser
     90B8   20 C312       4354          jsr  onedrv     ; select drive
     90BB   AD 0278       4355          lda  f2cnt
     90BE   48            4356          pha
     90BF   A9 01         4357          lda  #1
     90C1   8D 0278       4358          sta  f2cnt
     90C4   A9 FF         4359          lda  #$ff
     90C6   85 86         4360          sta  r0         ; set flag
     90C8   20 C44F       4361          jsr  lookup     ; locate file
     90CB   68            4362          pla
     90CC   8D 0278       4363          sta  f2cnt      ; restore var
     90CF   A5 37         4364          lda  fastsr
     90D1   29 7F         4365          and  #$7f       ; clr error recovery flag
     90D3   85 37         4366          sta  fastsr
     90D5   24 3B         4367          bit  switch     ; seq flag set ?
     90D7   30 06         4368          bmi  8$
                          4369
     90D9   A5 E7         4370          lda  pattyp     ; is it a program file ?
```

```
Error Addr  Code           Seq    Source statement

      90DB   C9 02        4371           cmp  #2
      90DD   D0 05        4372           bne  6$          ; not prg
                          4373
      90DF   AD 0280      4374   8$      lda  filtrk      ; check if found. err if not
      90E2   D0 08        4375           bne  1$          ; br, file found
                          4376
      90E4   A2 02        4377   6$      ldx  #%00000010 ; file not found
      90E6   2C           4378           .byte skip2
      90E7   A2 0F        4379   9$      ldx  #%00001111 ; no drive
      90E9   4C 91AD      4380           jmp  sys_err
                          4381
      90EC   85 7E        4382   1$      sta  prgtrk      ; save for next
      90EE   48           4383           pha              ; save track
      90EF   20 91DA      4384           jsr  set_buf     ; setup buffer allocation
      90F2   68           4385           pla              ; get track
      90F3   AE 02B0      4386           ldx  channel     ; get channel offset
      90F6   95 06        4387           sta  hdrs,x      ; setup track
      90F8   AD 0285      4388           lda  filsec      ; & sector
      90FB   8D 026F      4389           sta  prgsec      ; for next time
      90FE   95 07        4390           sta  hdrs+1,x
                          4391
      9100   A9 80        4392           lda  #read       ; read job
      9102   8D 0202      4393           sta  cmdbuf+2    ; save read cmd
      9105   85 5F        4394           sta  ctl_cmd
                          4395
      9107   58           4396   2$      cli              ; let controller run
      9108   A6 F9        4397           ldx  jobnum      ; get job #
      910A   A5 5F        4398           lda  ctl_cmd     ; get cmd
      910C   95 00        4399           sta  jobs,x      ; send cmd
      910E   20 864B      4400           jsr  stbctr      ; whack the controller in the head
      9111   E0 02        4401           cpx  #2          ; error ?
      9113   90 03        4402           bcc  5$
                          4403
      9115   4C 9199      4404           jmp  ctr_err
                          4405
      9118   78           4406   5$      sei
      9119   A0 00        4407           ldy  #0
      911B   B1 94        4408           lda  (dirbuf),y ; check status
      911D   F0 2F        4409           beq  end_of_file
                          4410
      911F   A5 37        4411           lda  fastsr      ; clear flag
      9121   29 FE        4412           and  #$fe
      9123   85 37        4413           sta  fastsr
                          4414
      9125   20 9228      4415           jsr  handsk      ; handshake error to the host
                          4416
      9128   A0 02        4417           ldy  #2
      912A   B1 94        4418   3$      lda  (dirbuf),y
      912C   AA           4419           tax              ; save data in .x
      912D   20 9228      4420           jsr  handsk      ; handshake it to the host
```

```
Error Addr  Code           Seq    Source statement

      9130  C8             4421           iny
      9131  D0 F7          4422           bne  3$
                           4423
      9133  AE 02B0        4424           ldx  channel     ; jobnum * 2
      9136  B1 94          4425           lda  (dirbuf),y  ; .y = 0
      9138  D5 06          4426           cmp  hdrs,x      ; same as previous track ?
      913A  F0 03          4427           beq  4$
                           4428
      913C  A0 80          4429           ldy  #read
      913E  2C             4430           .byte skip2
      913F  A0 88          4431   4$      ldy  #fread      ; fast read
      9141  84 5F          4432           sty  ctl_cmd     ; command to seek then read
      9143  95 06          4433           sta  hdrs,x      ; next track
      9145  A0 01          4434           ldy  #1          ; sector entry
      9147  B1 94          4435           lda  (dirbuf),y
      9149  95 07          4436           sta  hdrs+1,x    ; next sector
      914B  4C 9107        4437           jmp  2$
                           4438
                           4439
      914E                 4440   end_of_file
      914E  A2 1F          4441           ldx  #$1f        ; eof
      9150  20 9228        4442           jsr  handsk      ; handshake it to the host
                           4443
      9153  A9 01          4444           lda  #1
      9155  24 37          4445           bit  fastsr      ; first time through ?
      9157  F0 1E          4446           beq  1$          ; br, nope
                           4447
      9159  A8             4448           tay              ; .y = 1
      915A  B1 94          4449           lda  (dirbuf),y  ; number of bytes
      915C  38             4450           sec
      915D  E9 03          4451           sbc  #3
      915F  85 46          4452           sta  ctl_dat     ; save it
      9161  AA             4453           tax              ; send it
      9162  20 9228        4454           jsr  handsk      ; handshake it to the host
                           4455
      9165  C8             4456           iny              ; next
      9166  B1 94          4457           lda  (dirbuf),y  ; address low
      9168  AA             4458           tax
      9169  20 9228        4459           jsr  handsk      ; handshake it to the host
                           4460
      916C  C8             4461           iny
      916D  B1 94          4462           lda  (dirbuf),y  ; address high
      916F  AA             4463           tax
      9170  20 9228        4464           jsr  handsk      ; handshake it to the host
      9173  A0 04          4465           ldy  #4          ; skip addresses
      9175  D0 0D          4466           bne  3$          ; bra
                           4467
      9177  A0 01          4468   1$      ldy  #1
      9179  B1 94          4469           lda  (dirbuf),y  ; number of bytes
      917B  AA             4470           tax
```

```
Error Addr  Code           Seq    Source statement

     917C   CA             4471           dex
     917D   86 46          4472           stx  ctl_dat     ; save here
     917F   20 9228        4473           jsr  handsk      ; handshake it to the host
                           4474
     9182   A0 02          4475           ldy  #2          ; start at data
     9184   B1 94          4476   3$      lda  (dirbuf),y
     9186   AA             4477           tax
     9187   20 9228        4478           jsr  handsk      ; handshake it to the host
     918A   C8             4479           iny
     918B   C6 46          4480           dec  ctl_dat     ; use it as a temp
     918D   D0 F5          4481           bne  3$
                           4482
     918F   A9 00          4483           lda  #0
     9191   85 83          4484           sta  sa
     9193   20 DAC0        4485           jsr  close       ; close channel (faux)
     9196   4C C194        4486           jmp  endcmd
                           4487   ;
                           4488   ;
                           4489   ;
                           4490   ; *************************
                           4491   ; ***** ERROR HANDLER *****
                           4492   ; *************************
                           4493
     9199   78             4494   ctr_err sei              ; no irq's
     919A   86 46          4495           stx  ctl_dat     ; save status here
     919C   20 9228        4496           jsr  handsk      ; handshake it to the host
     919F   A9 00          4497           lda  #0
     91A1   85 83          4498           sta  sa
     91A3   20 DAC0        4499           jsr  close       ; close channel (faux)
     91A6   A6 F9          4500           ldx  jobnum
     91A8   A5 46          4501           lda  ctl_dat     ; get error
     91AA   4C E60A        4502           jmp  error       ; error out.....
                           4503
     91AD   78             4504   sys_err sei
     91AE   86 46          4505           stx  ctl_dat     ; save error
     91B0   A2 02          4506           ldx  #2          ; file not found
     91B2   20 9228        4507           jsr  handsk      ; give it to him
     91B5   A9 00          4508           lda  #0
     91B7   85 83          4509           sta  sa
     91B9   20 DAC0        4510           jsr  close       ; close channel (faux)
     91BC   A5 46          4511           lda  ctl_dat     ; get error back
     91BE   C9 02          4512           cmp  #2
     91C0   F0 03          4513           beq  1$
                           4514
     91C2   A9 74          4515           lda  #nodriv     ; no active drive
     91C4   2C             4516           .byte skip2
     91C5   A9 62          4517   1$      lda  #flntfd     ; file not found
     91C7   4C C1C8        4518           jmp  cmderr      ; never more...
                           4519   ;
                           4520   ;
```

```
Error Addr  Code           Seq   Source statement

                           4521  ;
                           4522  ; ************************************
                           4523  ; ***** FIND INTERNAL READ BUFFER *****
                           4524  ; ************************************
                           4525
     91CA   A9 00          4526  findbuf lda  #0
     91CC   85 83          4527          sta  sa          ; psydo-load
     91CE   A9 01          4528          lda  #1          ; 1 buffer
     91D0   20 D1E2        4529          jsr  getrch      ; find a read channel
     91D3   AA             4530          tax
     91D4   BD FEE0        4531          lda  bufind,x    ; get buffer
     91D7   85 95          4532          sta  dirbuf+1    ; set it up indirect
     91D9   60             4533          rts
                           4534  ;
                           4535  ;
                           4536  ;
                           4537  ; ************************************
                           4538  ; ***** SETUP INTERNAL READ BUFFER *****
                           4539  ; ************************************
                           4540
     91DA   A5 95          4541  set_buf lda  dirbuf+1   ; index to determine job
     91DC   38             4542          sec
     91DD   E9 03          4543          sbc  #3
     91DF   85 F9          4544          sta  jobnum      ; save in jobnum
     91E1   0A             4545          asl  a
     91E2   8D 02B0        4546          sta  channel     ; save channel off
     91E5   A9 00          4547          lda  #0
     91E7   85 94          4548          sta  dirbuf      ; even page boundary
     91E9   60             4549          rts
                           4550  ;
                           4551  ;
                           4552  ;
                           4553  ; ************************************
                           4554  ; ***** FAST LOAD FILENAME PARSER *****
                           4555  ; ************************************
                           4556
     91EA   A0 03          4557  set_fil ldy  #3          ; default .y
     91EC   AD 0274        4558          lda  cmdsiz      ; delete burst load command
     91EF   38             4559          sec
     91F0   E9 03          4560          sbc  #3
     91F2   8D 0274        4561          sta  cmdsiz      ; new command size
                           4562
     91F5   AD 0204        4563          lda  cmdbuf+4    ; drv # given ?
     91F8   C9 3A          4564          cmp  #':
     91FA   D0 0E          4565          bne  1$
                           4566
     91FC   AD 0203        4567          lda  cmdbuf+3
     91FF   AA             4568          tax              ; save
     9200   29 30          4569          and  #'0
     9202   C9 30          4570          cmp  #'0         ; 0:file ?
```

```
Error Addr  Code           Seq    Source statement

     9204   DO 04         4571           bne  1$
                          4572
     9206   E0 31         4573           cpx  #'1         ; chk for error
     9208   F0 1C         4574           beq  4$
                          4575
     920A   AD 0203       4576   1$      lda  cmdbuf+3    ; drv # given ?
     920D   C9 3A         4577           cmp  #':
     920F   D0 04         4578           bne  2$
                          4579
     9211   CE 0274       4580           dec  cmdsiz
     9214   C8            4581           iny
                          4582
     9215   A2 00         4583   2$      ldx  #0          ; start at cmdbuf+0
     9217   B9 0200       4584   3$      lda  cmdbuf,y    ; extract file-name
     921A   9D 0200       4585           sta  cmdbuf,x
     921D   C8            4586           iny
     921E   E8            4587           inx
     921F   EC 0274       4588           cpx  cmdsiz      ; done ?
     9222   D0 F3         4589           bne  3$          ; delete cmd from buffer
                          4590
     9224   18            4591           clc
     9225   24            4592           .byte skip1
     9226   38            4593   4$      sec              ; error
     9227   60            4594           rts
                          4595
                          4596
                          4597
     9228                 4598   handsk                   ; .x contains data
     9228   AD 1800       4599   1$      lda  pb          ; debounce
     922B   CD 1800       4600           cmp  pb
     922E   D0 F8         4601           bne  1$
                          4602
     9230   29 FF         4603           and  #$ff        ; set/clr neg flag
     9232   30 17         4604           bmi  3$          ; br, attn low
                          4605
     9234   45 37         4606           eor  fastsr      ; wait for state chg
     9236   29 04         4607           and  #4
     9238   F0 EE         4608           beq  1$
                          4609
     923A   8E 400C       4610           stx  sdr         ; send it
     923D   A5 37         4611           lda  fastsr
     923F   49 04         4612           eor  #4          ; change state of clk
     9241   85 37         4613           sta  fastsr
                          4614
     9243   A9 08         4615           lda  #8
     9245   2C 400D       4616   2$      bit  icr         ; wait transmission time
     9248   F0 FB         4617           beq  2$
                          4618
     924A   60            4619           rts
                          4620
```

```
Error Addr  Code           Seq    Source statement

     924B   4C A7B3       4621   3$       jmp  ptch30     ; bye-bye the host wants us
                          4622            .include  signature
```

```
Error Addr  Code           Seq    Source statement

                          4624            .subttl "signature.src'
                          4625
                          4626    ; test 256k-bit of rom
                          4627    ; using signature generation test bits 15,  11,  8,  6
                          4628    ; exit if ok
                          4629
     924E                 4630    signature
     924E   08            4631            php
     924F   78            4632            sei             ; no irqs
     9250   A2 00         4633            ldx  #0
     9252   86 00         4634            stx  $00        ; sig_lo
     9254   86 01         4635            stx  $01        ; sig_hi
     9256   A9 03         4636            lda  #3         ; skip checksum & signature bytes
     9258   85 75         4637            sta  ip         ; even page
     925A   A8            4638            tay
     925B   A9 80         4639            lda  #$80       ; start at $8000
     925D   85 76         4640            sta  ip+1       ; high order
     925F   B1 75         4641    2$      lda  (ip),y     ; get a byte
     9261   85 02         4642            sta  $02
     9263   A2 08         4643            ldx  #8         ; 8 bits in a byte right?
     9265   A5 02         4644    3$      lda  $02
     9267   29 01         4645            and  #1         ; bit 0
     9269   85 03         4646            sta  $03
     926B   A5 01         4647            lda  $01        ; get sig_hi
     926D   10 02         4648            bpl  4$         ; test bit 15
                          4649
     926F   E6 03         4650            inc  $03
     9271   6A            4651    4$      ror  a
     9272   90 02         4652            bcc  5$         ; test bit 8
                          4653
     9274   E6 03         4654            inc  $03
     9276   6A            4655    5$      ror  a
     9277   6A            4656            ror  a
     9278   6A            4657            ror  a
     9279   90 02         4658            bcc  6$
                          4659
     927B   E6 03         4660            inc  $03
     927D   A5 00         4661    6$      lda  $00        ; sig_lo
     927F   2A            4662            rol  a
     9280   2A            4663            rol  a
     9281   90 02         4664            bcc  7$         ; test bit 6
                          4665
     9283   E6 03         4666            inc  $03
     9285   66 03         4667    7$      ror  $03        ; sum into carry
     9287   26 00         4668            rol  $00        ; carry into bit 0 low byte
     9289   26 01         4669            rol  $01        ; carry into bit 0 high byte
                          4670
     928B   66 02         4671            ror  $02        ; ready for next bit
     928D   CA            4672            dex
     928E   D0 D5         4673            bne  3$
```

```
Error Addr  Code           Seq    Source statement

                          4674
     9290   E6 75         4675            inc  ip
     9292   DO CB         4676            bne  2$
                          4677
     9294   E6 76         4678            inc  ip+1       ; next page
     9296   DO C7         4679            bne  2$
                          4680
     9298   88            4681            dey
     9299   88            4682            dey
     929A   88            4683            dey             ; .y = 0
                          4684
     929B   A5 00         4685            lda  $00
     929D   CD 8000       4686            cmp  signature_lo
     92A0   DO 11         4687            bne  sig_err
                          4688
     92A2   A5 01         4689            lda  $01
     92A4   CD 8001       4690            cmp  signature_hi
     92A7   DO 0A         4691            bne  sig_err
                          4692
     92A9   84 00         4693            sty  $00        ; clear
     92AB   84 01         4694            sty  $01
     92AD   84 02         4695            sty  $02
     92AF   84 03         4696            sty  $03
                          4697
     92B1   28            4698            plp
     92B2   60            4699            rts
                          4700
                          4701
     92B3   A2 03         4702   sig_err  ldx  #3          ; 4 blinks
     92B5   86 6F         4703            stx  temp
     92B7   4C EA71       4704            jmp  perr       ; bye bye ....
                          4705            .include  gcrcntrl
```

```
Error Addr  Code           Seq    Source statement

                          4707            .subttl 'gcrcntrl.src'
                          4708
                          4709    ;     *contrl
                          4710    ;     main controller loop
                          4711    ;     scans job que for jobs
                          4712    ;     finds job on current track
                          4713    ;     if it exists
                          4714
     92BA   BA            4715    jlcc    tsx                 ; save current stack pointer
     92BB   86 49         4716            stx  savsp          ; *
                          4717
     92BD   2C 1C04       4718            bit  tilc2          ; reset irq flag
     92C0   AD 1C0C       4719            lda  pcr2           ; enable SO
     92C3   09 0E         4720            ora  #$0e
     92C5   8D 1C0C       4721            sta  pcr2
                          4722
     92C8   A0 05         4723    jtop    ldy  #numjob-1      ; pointer into job que
     92CA   B9 0000       4724    2$      lda  jobs,y         ; find a job (msb set)
     92CD   30 06         4725            bmi  3$             ; branch if there is a job
     92CF   88            4726            dey
     92D0   10 F8         4727            bpl  2$
                          4728
     92D2   4C 99CA       4729            jmp  jend
                          4730
     92D5   C9 88         4731    3$      cmp  #fread         ; fast read cmd
     92D7   D0 03         4732            bne  1$
                          4733
     92D9   4C 960D       4734            jmp  jread01        ; same track read job
                          4735
     92DC   C9 D0         4736    1$      cmp  #jumpc         ; jump cmd ?
     92DE   D0 03         4737            bne  15$
                          4738
     92E0   4C 93A2       4739            jmp  exdo
                          4740
     92E3   29 01         4741    15$     and  #drvmsk        ; get drive #
     92E5   F0 07         4742            beq  5$
                          4743
     92E7   84 3F         4744            sty  jobn
     92E9   A9 0F         4745            lda  #15
     92EB   4C 99B5       4746            jmp  jerrr          ; bad drv
                          4747
     92EE   AA            4748    5$      tax
     92EF   C5 3E         4749            cmp  cdrive         ; test if current drive
     92F1   F0 08         4750            beq  7$
                          4751
     92F3   85 3E         4752            sta  cdrive
     92F5   20 F97E       4753            jsr  turnon         ; turn drive on
     92F8   4C 99CA       4754            jmp  jend           ; go clean up
                          4755
     92FB   A5 20         4756    7$      lda  drvst          ; test if motor up to speed
```

```
Error Addr  Code           Seq    Source statement

     92FD   30 03         4757            bmi  8$
                          4758
     92FF   0A             4759            asl  a           ; test if stepping
     9300   10 03         4760            bpl  9$          ; not stepping
                          4761
     9302   4C 99CA       4762   8$       jmp  jend
                          4763
     9305   A9 20         4764   9$       lda  #$20        ; status=running
     9307   85 20         4765            sta  drvst
                          4766
     9309   A0 05         4767            ldy  #numjob-1
     930B   84 3F         4768            sty  jobn
                          4769
     930D   20 93D1       4770   10$      jsr  jsetjb
     9310   30 1A         4771            bmi  11$
                          4772
     9312   C6 3F         4773   12$      dec  jobn
     9314   10 F7         4774            bpl  10$
                          4775
     9316   A4 41         4776            ldy  nxtjob
     9318   20 93D3       4777            jsr  jsetjb1
                          4778
     931B   A5 42         4779            lda  nxtrk
     931D   85 4A         4780            sta  steps
     931F   06 4A         4781            asl  steps       ; steps*2
                          4782
     9321   A9 60         4783            lda  #$60        ; set status=stepping
     9323   85 20         4784            sta  drvst
                          4785
     9325   B1 32         4786            lda  (hdrpnt),y ; get dest track #
     9327   85 22         4787            sta  drvtrk
     9329   4C 99CA       4788            jmp  jend
                          4789
     932C   29 01         4790   11$      and  #drvmsk     ; test if same drive
     932E   C5 3E         4791            cmp  cdrive
     9330   D0 E0         4792            bne  12$         ; test if current drive
                          4793
     9332   A5 22         4794            lda  drvtrk
     9334   F0 32         4795            beq  13$         ; uninit. track #
                          4796
     9336   A5 22         4797            lda  drvtrk      ; we are here
     9338   C9 24         4798            cmp  #36         ; destination
     933A   08             4799            php              ; save status
     933B   B1 32         4800            lda  (hdrpnt),y ; destination
     933D   C9 24         4801            cmp  #36
     933F   6A             4802            ror  a           ; rotate to sign bit
     9340   28             4803            plp              ; retrieve carry
     9341   29 80         4804            and  #$80        ; retrieve sign
     9343   90 0B         4805            bcc  16$         ; br, we are on side 0
                          4806
```

```
Error Addr  Code          Seq   Source statement

     9345   30 11        4807           bmi  17$         ; br, we are on side 1 & our dest is side 1
                         4808
                         4809   ; a change ...
                         4810
     9347   A5 22        4811           lda  drvtrk      ; we are on side 1 & and our dest is side 0
     9349   E9 23        4812           sbc  #35         ; so decrement where we are
     934B   85 22        4813           sta  drvtrk
     934D   4C 9358      4814           jmp  17$
                         4815
     9350   10 06        4816   16$     bpl  17$         ; br, we are on side 0 & our dest is side 0
                         4817
                         4818   ; a change ...
                         4819
     9352   A5 22        4820           lda  drvtrk      ; we are on side 0 & our dest is side 1
     9354   69 23        4821           adc  #35
     9356   85 22        4822           sta  drvtrk
     9358   38           4823   17$     sec
     9359   B1 32        4824           lda  (hdrpnt),y
     935B   E5 22        4825           sbc  drvtrk
     935D   F0 09        4826           beq  13$         ; on track
                         4827
     935F   85 42        4828           sta  nxtrk
     9361   A5 3F        4829           lda  jobn        ; save job# and dist to track
     9363   85 41        4830           sta  nxtjob
                         4831
     9365   4C 9312      4832           jmp  12$
                         4833
                         4834
     9368   A2 04        4835   13$     ldx  #4          ; set track and sector
     936A   B1 32        4836           lda  (hdrpnt),y
     936C   85 40        4837           sta  tracc
     936E   C9 24        4838           cmp  #36         ; second side ?
     9370   A8           4839           tay              ; save track
     9371   20 93F3      4840           jsr  set_side    ; adjust h/w and set appropiate flag
     9374   98           4841           tya              ; restore track
     9375   90 02        4842           bcc  14$
                         4843
     9377   E9 23        4844           sbc  #35         ; track 36 is track 1 physically
                         4845
     9379   AA           4846   14$     tax
     937A   BD 9408      4847           lda  worktbl-1,x
     937D   85 43        4848           sta  sectr       ; select density
                         4849
     937F   AD 1C00      4850           lda  dskcnt
     9382   29 9F        4851           and  #$9f        ; clear density bits
     9384   05 43        4852           ora  sectr
     9386   8D 1C00      4853           sta  dskcnt      ; set density bits
                         4854
     9389   BD 942B      4855           lda  num_sec-1,x
     938C   85 43        4856           sta  sectr       ; get # of sector
```

```
Error Addr  Code           Seq    Source statement

                          4857
      938E   A5 45        4858            lda  job         ; yes, go do the job
      9390   C9 40        4859            cmp  #bump-$80  ; test for bump
      9392   F0 1C        4860            beq  jbmp
                          4861
      9394   C9 60        4862            cmp  #execd
      9396   F0 0A        4863            beq  exdo
                          4864
      9398   C9 70        4865            cmp  #frmt-$80  ; is it a format code
      939A   F0 03        4866            beq  jfmt
                          4867
      939C   4C 944F      4868            jmp  jseak      ; do a sector seek
                          4869
      939F   4C 9B29      4870  jfmt      jmp  jformt     ; do the format
                          4871
      93A2   A5 3F        4872  exdo      lda  jobn
      93A4   18           4873            clc
      93A5   69 03        4874            adc  #>bufs
      93A7   85 31        4875            sta  bufpnt+1
      93A9   A9 00        4876            lda  #0
      93AB   85 30        4877            sta  bufpnt
      93AD   6C 0030      4878            jmp  (bufpnt)
                          4879
      93B0                4880  jbmp
      93B0   A9 60        4881            lda  #$60       ; set status=stepping
      93B2   85 20        4882            sta  drvst
      93B4   AD 1C00      4883            lda  dskcnt
      93B7   29 FC        4884            and  #$ff-$03   ; set phase a
      93B9   8D 1C00      4885            sta  dskcnt
                          4886
      93BC   A9 A4        4887            lda  #256-92    ; step back 45 traks
      93BE   85 4A        4888            sta  steps
      93C0   AD 01B1      4889            lda  side       ; what side are we on ?
      93C3   30 03        4890            bmi  1$
                          4891
      93C5   A9 01        4892            lda  #$01       ; drvtrk now 1
      93C7   2C           4893            .byte skip2
      93C8   A9 24        4894  1$        lda  #36        ; drvtrk now 36
      93CA   85 22        4895            sta  drvtrk
      93CC   A9 01        4896            lda  #1
      93CE   4C 99B5      4897            jmp  jerrr
                          4898
      93D1   A4 3F        4899  jsetjb    ldy  jobn
      93D3   B9 0000      4900  jsetjb1   lda  jobs,y
      93D6   48           4901            pha
      93D7   10 14        4902            bpl  1$         ; no job here
                          4903
      93D9   29 78        4904            and  #$78
      93DB   85 45        4905            sta  job
      93DD   98           4906            tya
```

```
Error Addr  Code           Seq    Source statement

     93DE   0A            4907            asl  a
     93DF   69 06         4908            adc  #<hdrs      ; hdrpnt,trk,sec for job
     93E1   85 32         4909            sta  hdrpnt
     93E3   A9 00         4910            lda  #>hdrs
     93E5   85 33         4911            sta  hdrpnt+1
     93E7   98            4912            tya              ; point at buffer
     93E8   18            4913            clc
     93E9   69 03         4914            adc  #>bufs
     93EB   85 31         4915            sta  bufpnt+1
                          4916
     93ED   A0 00         4917    1$      ldy  #0
     93EF   84 30         4918            sty  bufpnt
     93F1   68            4919            pla
     93F2   60            4920            rts
                          4921
                          4922
     93F3                 4923    set_side
     93F3   B0 03         4924            bcs  1$
     93F5   A9 00         4925            lda  #0
     93F7   2C            4926            .byte  skip2
     93F8   A9 84         4927    1$      lda  #$84
     93FA   8D 01B1       4928            sta  side
     93FD   AD 180F       4929            lda  pota1
     9400   29 FB         4930            and  #$ff-4
     9402   0D 01B1       4931            ora  side
     9405   8D 180F       4932            sta  pota1
     9408   60            4933            rts
                          4934
                          4935
     9409                 4936    worktbl ; table of track densities
     9409   60 60 60      4937            .byte $60,$60,$60,$60,$60,$60,$60,$60,$60,$60
     940C   60 60 60
     940F   60 60 60
     9412   60
     9413   60 60 60      4938            .byte $60,$60,$60,$60,$60,$60,$60
     9416   60 60 60
     9419   60
     941A   40 40 40      4939            .byte $40,$40,$40,$40,$40,$40,$40
     941D   40 40 40
     9420   40
     9421   20 20 20      4940            .byte $20,$20,$20,$20,$20,$20
     9424   20 20 20
     9427   00 00 00      4941            .byte $00,$00,$00,$00,$00
     942A   00 00
                          4942
     942C                 4943    num_sec ; number of sectors per track
     942C   15 15 15      4944            .byte 21,21,21,21,21,21,21,21,21,21
     942F   15 15 15
     9432   15 15 15
     9435   15
```

```
Error Addr  Code           Seq    Source statement

     9436   15 15 15      4945           .byte 21,21,21,21,21,21,21
     9439   15 15 15
     943C   15
     943D   13 13 13      4946           .byte 19,19,19,19,19,19,19
     9440   13 13 13
     9443   13
     9444   12 12 12      4947           .byte 18,18,18,18,18,18
     9447   12 12 12
     944A   11 11 11      4948           .byte 17,17,17,17,17
     944D   11 11
                          4949           .include  lccseek1
```

```
Error Addr  Code           Seq    Source statement

                          4951           .subttl 'lccseek1.src'
                          4952
     944F   A9 5A         4953   jseak   lda  #90         ; search 90 headers
     9451   85 4B         4954           sta  tmp
                          4955
     9453   20 9754       4956   1$      jsr  jsync       ; find sync char
                          4957
     9456   2C 180F       4958   2$      bit  pota1       ; wait for block id
     9459   30 FB         4959           bmi  2$
                          4960
     945B   AD 1C01       4961           lda  data2       ; clear pa1 in the gate array
     945E   C9 52          4962           cmp  #$52        ; test if header block
     9460   D0 3E         4963           bne  3$          ; not header
                          4964
     9462   99 0024       4965           sta  stab,y      ; store 1st byte
     9465   C8            4966           iny
                          4967
     9466   2C 180F       4968   4$      bit  pota1
     9469   30 FB         4969           bmi  4$
                          4970
     946B   AD 1C01       4971           lda  data2
     946E   99 0024       4972           sta  stab,y      ; store gcr header off
                          4973
     9471   C8            4974           iny
     9472   C0 08         4975           cpy  #8          ; 8 gcr bytes in header
     9474   D0 F0         4976           bne  4$
                          4977
     9476   20 952F       4978           jsr  jcnvbin     ; convert header in stabof to binary in header
                          4979
     9479   A0 04         4980           ldy  #4          ; compute checksum
     947B   A9 00          4981           lda  #0
                          4982
     947D   59 0016       4983   5$      eor  header,y
     9480   88            4984           dey
     9481   10 FA         4985           bpl  5$
                          4986
     9483   C9 00         4987           cmp  #0          ; test if ok
     9485   D0 30         4988           bne  9$          ; nope, checksum error in header
                          4989
     9487   A5 18         4990           lda  header+2
     9489   85 22         4991           sta  drvtrk
                          4992
     948B   A5 45         4993           lda  job         ; test if a seek job
     948D   C9 30         4994           cmp  #$30
     948F   F0 18         4995           beq  6$
                          4996
     9491   A5 12         4997           lda  dskid
     9493   C5 16         4998           cmp  header
     9495   D0 1D         4999           bne  8$
                          5000
```

```
Error Addr  Code           Seq    Source statement

     9497   A5 13         5001           lda  dskid+1
     9499   C5 17         5002           cmp  header+1
     949B   DO 17         5003           bne  8$
                          5004
     949D   4C 94BC       5005           jmp  7$          ; find best sector to service
                          5006
     94AO   C6 4B         5007   3$      dec  tmp         ; search more?
     94A2   DO AF         5008           bne  1$          ; yes
                          5009
     94A4   A9 O2         5010           lda  #2          ; cant find a sector
     94A6   20 99B5       5011           jsr  jerrr
                          5012
     94A9   A5 16         5013   6$      lda  header      ; sta disk id's
     94AB   85 12         5014           sta  dskid       ; *
     94AD   A5 17         5015           lda  header+1
     94AF   85 13         5016           sta  dskid+1
                          5017
     94B1   A9 O1         5018           lda  #1          ; return ok code
     94B3   2C            5019           .byte    skip2
                          5020
     94B4   A9 OB         5021   8$      lda  #11         ; disk id mismatch
     94B6   2C            5022           .byte    skip2
                          5023
     94B7   A9 O9         5024   9$      lda  #9          ; checksum error in header
     94B9   4C 99B5       5025           jmp  jerrr
                          5026
     94BC   A9 7F         5027   7$      lda  #$7f        ; find best job
     94BE   85 4C         5028           sta  csect
                          5029
     94CO   A5 19         5030           lda  header+3    ; get upcoming sector #
     94C2   18            5031           clc
     94C3   69 O2         5032           adc  #2
     94C5   C5 43         5033           cmp  sectr
     94C7   90 O2         5034           bcc  10$
                          5035
     94C9   E5 43         5036           sbc  sectr       ; wrap around
                          5037
     94CB   85 4D         5038   10$     sta  nexts       ; next sector
                          5039
     94CD   A2 O5         5040           ldx  #numjob-1
     94CF   86 3F         5041           stx  jobn
                          5042
     94D1   A2 FF         5043           ldx  #$ff
                          5044
     94D3   20 93D1       5045   12$     jsr  jsetjb
     94D6   10 43         5046           bpl  11$
                          5047
     94D8   29 O1         5048           and  #drvmsk
     94DA   C5 3E         5049           cmp  cdrive      ; test if same drive
     94DC   DO 3D         5050           bne  11$         ; nope
```

```
Error Addr  Code           Seq    Source statement

                          5051
      94DE   A0 00         5052            ldy  #0          ; test if same track
      94E0   B1 32         5053            lda  (hdrpnt),y
      94E2   C5 40         5054            cmp  tracc
      94E4   D0 35         5055            bne  11$
                          5056
      94E6   A5 45         5057            lda  job
      94E8   C9 60         5058            cmp  #execd
      94EA   F0 0C         5059            beq  13$
                          5060
      94EC   A0 01         5061            ldy  #1
      94EE   38            5062            sec
      94EF   B1 32         5063            lda  (hdrpnt),y
      94F1   E5 4D         5064            sbc  nexts
      94F3   10 03         5065            bpl  13$
                          5066
      94F5   18            5067            clc
      94F6   65 43         5068            adc  sectr
                          5069
      94F8   C5 4C         5070   13$      cmp  csect
      94FA   B0 1F         5071            bcs  11$
                          5072
      94FC   48            5073            pha              ; save it
      94FD   A5 45         5074            lda  job
      94FF   F0 15         5075            beq  16$         ; must be a read
                          5076
      9501   68            5077            pla
      9502   C9 09         5078            cmp  #wrtmin     ;  if(csect<9)return;
      9504   90 15         5079            bcc  11$         ;  if(csect>12)return;
                          5080
      9506   C9 0C         5081            cmp  #wrtmax
      9508   B0 11         5082            bcs  11$
                          5083
      950A   85 4C         5084   15$      sta  csect       ; its better
      950C   A5 3F         5085            lda  jobn
      950E   AA            5086            tax
      950F   18            5087            clc
      9510   69 03         5088            adc  #>bufs
      9512   85 31         5089            sta  bufpnt+1
                          5090
      9514   D0 05         5091            bne  11$
                          5092
      9516   68            5093   16$      pla
      9517   C9 06         5094            cmp  #rdmax      ; if(csect>6)return;
      9519   90 EF         5095            bcc  15$
                          5096
      951B   C6 3F         5097   11$      dec  jobn
      951D   10 B4         5098            bpl  12$
                          5099
      951F   8A            5100            txa              ; test if a job to do
```

```
Error Addr  Code           Seq    Source statement

     9520   10 03         5101            bpl   14$
                          5102
     9522   4C 99CA       5103            jmp   jend        ; no job found
                          5104
     9525   86 3F         5105    14$     stx   jobn
     9527   20 93D1       5106            jsr   jsetjb
     952A   A5 45         5107            lda   job
     952C   4C 9606       5108            jmp   jread
                          5109
     952F   A5 30         5110    jcnvbin lda   bufpnt
     9531   48            5111            pha
     9532   A5 31         5112            lda   bufpnt+1
     9534   48            5113            pha               ; save buffer pntr
                          5114
     9535   A9 24         5115            lda   #<stab      ; stab offset
     9537   85 30         5116            sta   bufpnt      ; point at gcr code
     9539   A9 00         5117            lda   #>stab
     953B   85 31         5118            sta   bufpnt+1
                          5119
     953D   A9 00         5120            lda   #0
     953F   85 34         5121            sta   gcrpnt
                          5122
     9541   20 98D9       5123            jsr   jget4gb     ; convert 4 bytes
                          5124
     9544   A5 55         5125            lda   btab+3
     9546   85 18         5126            sta   header+2
                          5127
     9548   A5 54         5128            lda   btab+2
     954A   85 19         5129            sta   header+3
                          5130
     954C   A5 53         5131            lda   btab+1
     954E   85 1A         5132            sta   header+4
                          5133
                          5134
     9550   20 98D9       5135            jsr   jget4gb     ; get 2 more
                          5136
     9553   A5 52         5137            lda   btab        ; get id
     9555   85 17         5138            sta   header+1
     9557   A5 53         5139            lda   btab+1
     9559   85 16         5140            sta   header
                          5141
     955B   68            5142            pla
     955C   85 31         5143            sta   bufpnt+1    ; restore pointer
     955E   68            5144            pla
     955F   85 30         5145            sta   bufpnt
     9561   60            5146            rts
                          5147            .include  lccread1
```

```
Error Addr  Code           Seq   Source statement

                          5149           .subttl 'lccread1.src'
                          5150
                          5151   ;=============================================================
                          5152   ;=     8     =     8     =     8     =     8     =     8     =
                          5153   ;=============================================================
                          5154   ;   (5) (3  + 2)  (5) (1+  4)   (4  +1) (5)  (2 +  3)   (5)
                          5155   ;    a      b      c    d          e    f      g       h
                          5156   ;=============================================================
                          5157
                          5158   ;    read in track,sector specified
                          5159   ;    in header
                          5160
          =9600           5161           *=*+256-<*      ;  even page
                          5162
     9600   20 970F       5163   jdstrt  jsr  jsrch      ;  find header
     9603   4C 9754       5164           jmp  jsync      ;  and then data block sync
                          5165
     9606   C9 00         5166   jread   cmp  #0         ;  test if read job
     9608   F0 03         5167           beq  jread01    ;  go test if write
     960A   4C 976E       5168           jmp  jwright
                          5169
     960D   20 9600       5170   jread01 jsr  jdstrt     ;  find header and start reading data
                          5171                           ; sync routine sets y to 0
                          5172
     9610   2C 180F       5173   1$      bit  pota1      ;  4
     9613   30 FB         5174           bmi  1$         ;  3 + 2
                          5175
     9615   AD 1C01       5176           lda  data2      ;  4
     9618   AA            5177           tax             ;  2   reg x = xxxxx000
     9619   BD A00D       5178           lda  gcrtb1,x   ;  4   nibble a
     961C   85 52         5179           sta  btab       ;  3
     961E   8A            5180           txa             ;  2
     961F   29 07         5181           and  #%00000111 ;  2
     9621   85 53         5182           sta  btab+1     ;  3   extract 3 bits nibble b
                          5183
     9623   2C 180F       5184   2$      bit  pota1      ;  4
     9626   30 FB         5185           bmi  2$         ;  3 + 2
                          5186
     9628   AD 1C01       5187           lda  data2      ;  4
     962B   85 54         5188           sta  btab+2     ;  3
     962D   29 C0         5189           and  #%11000000 ;  2   extract 2 bits nibble b
     962F   05 53         5190           ora  btab+1     ;  3
     9631   AA            5191           tax             ;  2   reg x = xx000xxx
     9632   BD 9F0D       5192           lda  gcrtba,x   ;  4   nibble b
     9635   05 52         5193           ora  btab       ;  3
     9637   48            5194           pha             ;  3
     9638   4C 9667       5195           jmp  5$         ;  3
                          5196
                          5197   ;*****************************************************************
                          5198
```

```
Error Addr  Code           Seq    Source statement

     963B   2C 180F       5199   3$      bit  pota1       ;  4
     963E   30 FB          5200          bmi  3$          ;  3 + 2
                           5201
     9640   AD 1C01       5202          lda  data2       ;  4
     9643   AA            5203          tax              ;  2   reg x = xxxxx000
     9644   BD A00D       5204          lda  gcrtb1,x    ;  4   nibble a
     9647   85 52         5205          sta  btab        ;  3
     9649   8A            5206          txa              ;  2
     964A   29 07         5207          and  #%00000111  ;  2
     964C   85 53         5208          sta  btab+1      ;  3   extract 3 bits nibble b
                           5209
     964E   2C 180F       5210   4$      bit  pota1       ;  4
     9651   30 FB          5211          bmi  4$          ;  3 + 2
                           5212
     9653   AD 1C01       5213          lda  data2       ;  4
     9656   85 54         5214          sta  btab+2      ;  3
     9658   29 C0         5215          and  #%11000000  ;  2
     965A   05 53         5216          ora  btab+1      ;  3
     965C   AA            5217          tax              ;  2   reg x = xx000xxx
     965D   BD 9F0D       5218          lda  gcrtba,x    ;  4   nibble b
     9660   05 52         5219          ora  btab        ;  3
     9662   91 30         5220          sta  (bufpnt),y  ;  6
     9664   C8            5221          iny              ;  2
     9665   F0 70         5222          beq  6$          ;  2
                           5223
     9667   A5 54         5224   5$      lda  btab+2      ;  3
     9669   AA            5225          tax              ;  2   reg x = 00xxxxx0
     966A   BD A10D       5226          lda  gcrtb2,x    ;  4   nibble c
     966D   85 52         5227          sta  btab        ;  3
     966F   8A            5228          txa              ;  2
     9670   29 01         5229          and  #%00000001  ;  2
     9672   85 54         5230          sta  btab+2      ;  3   extract 1 bits nibble d
                           5231
     9674   2C 180F       5232   7$      bit  pota1       ;  4
     9677   30 FB          5233          bmi  7$          ;  3 + 2
                           5234
     9679   AD 1C01       5235          lda  data2       ;  4
     967C   85 55         5236          sta  btab+3      ;  3
     967E   29 F0         5237          and  #%11110000  ;  2
     9680   05 54         5238          ora  btab+2      ;  3
     9682   AA            5239          tax              ;  2   reg x = xxxx000x
     9683   BD 9F0F       5240          lda  gcrtbd,x    ;  4   nibble d
     9686   05 52         5241          ora  btab        ;  3
     9688   91 30         5242          sta  (bufpnt),y  ;  6
     968A   C8            5243          iny              ;  2
     968B   A5 55         5244          lda  btab+3      ;  3
     968D   29 0F         5245          and  #%00001111  ;  2
     968F   85 55         5246          sta  btab+3      ;  3   extract 4 bits nibble e
                           5247
     9691   2C 180F       5248   8$      bit  pota1       ;  4
```

```
Error Addr  Code           Seq    Source statement

     9694   30 FB         5249            bmi  8$          ;  3 + 2
                          5250
     9696   AD 1C01       5251            lda  data2       ;  4
     9699   85 3A         5252            sta  chksum      ;  3
     969B   29 80         5253            and  #%10000000  ;  2
     969D   05 55         5254            ora  btab+3      ;  3
     969F   AA            5255            tax              ;  2   reg x = x000xxxx
     96A0   BD 9F1D       5256            lda  gcrtbe,x    ;  4   nibble e
     96A3   85 52         5257            sta  btab        ;  3
     96A5   A5 3A         5258            lda  chksum      ;  3
     96A7   AA            5259            tax              ;  2   reg x = 0xxxxx00
     96A8   BD A20D       5260            lda  gcrtb3,x    ;  4   nibble f
     96AB   05 52         5261            ora  btab        ;  3
     96AD   91 30         5262            sta  (bufpnt),y  ;  6
     96AF   C8            5263            iny              ;  2
     96B0   8A            5264            txa              ;  2
     96B1   29 03         5265            and  #%00000011  ;  2
     96B3   85 3A         5266            sta  chksum      ;  3   extract 2 bits nibble g
                          5267
     96B5   2C 180F       5268   9$       bit  pota1       ;  4
     96B8   30 FB         5269            bmi  9$          ;  3 + 2
                          5270
     96BA   AD 1C01       5271            lda  data2       ;  4
     96BD   85 53         5272            sta  btab+1      ;  3
     96BF   29 E0         5273            and  #%11100000  ;  2
     96C1   05 3A         5274            ora  chksum      ;  3
     96C3   AA            5275            tax              ;  2   reg x = xxx000xx
     96C4   BD 9F2A       5276            lda  gcrtbg,x    ;  4   nibble g
     96C7   85 52         5277            sta  btab        ;  3
     96C9   A5 53         5278            lda  btab+1      ;  3
     96CB   AA            5279            tax              ;  2   reg x = 000xxxxx
     96CC   BD A30D       5280            lda  gcrtb4,x    ;  4   nibble h
     96CF   05 52         5281            ora  btab        ;  3
     96D1   91 30         5282            sta  (bufpnt),y  ;  6
     96D3   C8            5283            iny              ;  2
     96D4   4C 963B       5284            jmp  3$          ;  4
                          5285
                          5286   ;*************************************************************************
                          5287
     96D7   A5 54         5288   6$       lda  btab+2      ;  3
     96D9   AA            5289            tax              ;  2   reg x = 00xxxxx0
     96DA   BD A10D       5290            lda  gcrtb2,x    ;  4   nibble c
     96DD   85 52         5291            sta  btab        ;  3
     96DF   8A            5292            txa              ;  2
     96E0   29 01         5293            and  #%00000001  ;  2
     96E2   85 54         5294            sta  btab+2      ;  3
                          5295
     96E4   2C 180F       5296   10$      bit  pota1       ;  4
     96E7   30 FB         5297            bmi  10$         ;  3 + 2
                          5298
```

```
Error Addr  Code           Seq    Source statement

     96E9   AD 1C01       5299           lda  data2       ;  4
     96EC   29 F0         5300           and  #%11110000 ;  2
     96EE   05 54         5301           ora  btab+2     ;  3
     96F0   AA            5302           tax            ;  2   reg x = xxxx000x
     96F1   BD 9F0F       5303           lda  gcrtbd,x   ;  4   nibble d
     96F4   05 52         5304           ora  btab       ;  3
     96F6   85 53         5305           sta  btab+1     ;  3   store off cs byte
                          5306
     96F8   68            5307           pla            ; retrieve first byte off of disk
     96F9   C5 47         5308           cmp  dbid       ; see if it is a 7
     96FB   D0 0A         5309           bne  12$        ; br, nope
                          5310
                          5311
     96FD   20 F5E9       5312           jsr  chkblk     ; calc checksum
     9700   C5 53         5313           cmp  btab+1
     9702   F0 06         5314           beq  11$
                          5315
     9704   A9 05         5316           lda  #5         ; data block checksum error
     9706   2C            5317           .byte    skip2
     9707   A9 04         5318   12$     lda  #4
     9709   2C            5319           .byte    skip2
                          5320
     970A   A9 01         5321   11$     lda  #1         ; read data block ok
     970C   4C 99B5       5322           jmp  jerrr
                          5323
                          5324
     970F   A5 12         5325   jsrch   lda  dskid      ; get master id for the drive
     9711   85 16         5326           sta  header
     9713   A5 13         5327           lda  dskid+1
     9715   85 17         5328           sta  header+1
                          5329
     9717   A0 00         5330           ldy  #0         ; get track,sectr
     9719   B1 32         5331           lda  (hdrpnt),y
     971B   85 18         5332           sta  header+2
     971D   C8            5333           iny
     971E   B1 32         5334           lda  (hdrpnt),y
     9720   85 19         5335           sta  header+3
                          5336
     9722   A9 00         5337           lda  #0
     9724   45 16         5338           eor  header
     9726   45 17         5339           eor  header+1
     9728   45 18         5340           eor  header+2
     972A   45 19         5341           eor  header+3
     972C   85 1A         5342           sta  header+4   ; store the checksum
     972E   20 F934       5343           jsr  conhdr     ; convert header to gcr
                          5344
     9731   A9 5A         5345           lda  #90        ; search 90 sync chars
     9733   85 4B         5346           sta  tmp
     9735   20 9754       5347   1$      jsr  jsync      ; find sync
                          5348
```

```
Error Addr  Code           Seq    Source statement

     9738   B9 0024       5349   2$       lda  stab,y      ; what it should be
     973B   2C 180F       5350   3$       bit  pota1
     973E   30 FB         5351            bmi  3$
                          5352
     9740   CD 1C01       5353            cmp  data2       ; is it the same .cmp absolute
     9743   D0 06         5354            bne  4$          ; nope
                          5355
     9745   C8            5356            iny
     9746   C0 08         5357            cpy  #8
     9748   D0 EE         5358            bne  2$
                          5359
     974A   60            5360            rts
                          5361
     974B   C6 4B         5362   4$       dec  tmp         ; try again
     974D   D0 E6         5363            bne  1$
                          5364
     974F   A9 02         5365            lda  #2          ; cant find this header
     9751   4C 99B5       5366            jmp  jerrr
                          5367
                          5368
     9754   A2 0F         5369   jsync    ldx  #15
     9756   A0 00         5370            ldy  #0          ; s/w timers ok
     9758   2C 1C00       5371   1$       bit  dskcnt      ; sync a synch ?
     975B   10 0B         5372            bpl   2$
                          5373
     975D   88            5374            dey
     975E   D0 F8         5375            bne  1$
                          5376
     9760   CA            5377            dex
     9761   D0 F5         5378            bne  1$
                          5379
     9763   A9 03         5380            lda  #3
     9765   4C 99B5       5381            jmp  jerrr       ; sync error
                          5382
     9768   AD 1C01       5383   2$       lda  data2       ; clear pa latch
     976B   A0 00         5384            ldy  #0          ; clear pointer
     976D   60            5385            rts
                          5386            .include  lccwrt1
```

```
Error Addr  Code           Seq   Source statement

                          5388           .subttl 'lccwrt1.src'
                          5389
                          5390   ;    write out data buffer
                          5391
     976E   C9 10         5392   jwright cmp  #$10        ;  test if write
     9770   F0 03         5393           beq  1$
                          5394
     9772   4C 9898       5395           jmp  jvrfy
                          5396
     9775   20 F5E9       5397   1$      jsr  chkblk      ;  get block checksum
     9778   85 3A         5398           sta  chksum
                          5399
     977A   AD 1C00       5400           lda  dskcnt      ;  test for write protect
     977D   29 10         5401           and  #$10
     977F   D0 05         5402           bne  2$          ;  not  protected
                          5403
     9781   A9 08         5404           lda  #8          ;  write protect error
     9783   4C 99B5       5405           jmp  jerrr
                          5406
     9786   20 F78F       5407   2$      jsr  bingcr      ;  convert buffer to write image
                          5408
     9789   20 970F       5409           jsr  jsrch       ;  find header
                          5410
     978C   A0 09         5411           ldy  #gap1-2     ;  wait out header gap
                          5412
     978E   2C 180F       5413   3$      bit  pota1
     9791   30 FB         5414           bmi  3$
                          5415
     9793   2C 1C00       5416           bit  byt_clr
                          5417
     9796   88            5418           dey              ;  test if done yet
     9797   D0 F5         5419           bne  3$
                          5420
     9799   A9 FF         5421           lda  #$ff        ;  make output $ff
     979B   8D 1C03       5422           sta  ddra2
                          5423
     979E   AD 1C0C       5424           lda  pcr2        ;  set write mode
     97A1   29 1F         5425           and  #$ff-$e0    ;  0=wr
     97A3   09 C0         5426           ora  #$c0
     97A5   8D 1C0C       5427           sta  pcr2
                          5428
     97A8   A9 FF         5429           lda  #$ff        ;  write 4 gcr sync
     97AA   A0 05         5430           ldy  #numsyn
     97AC   8D 1C01       5431           sta  data2
                          5432
     97AF   2C 180F       5433   4$      bit  pota1
     97B2   30 FB         5434           bmi  4$
                          5435
     97B4   2C 1C00       5436           bit  byt_clr
                          5437
```

```
Error Addr  Code           Seq    Source statement

     97B7   88             5438            dey
     97B8   D0 F5           5439            bne  4$
                           5440
                           5441    ; write out overflow buffer
                           5442
     97BA   A0 BB           5443            ldy  #256-topwrt
                           5444
     97BC   B9 0100         5445    5$      lda  ovrbuf,y    ; get a char
     97BF   2C 180F         5446    6$      bit  pota1
     97C2   30 FB           5447            bmi  6$
                           5448
     97C4   8D 1C01         5449            sta  data2       ;  stuff it
                           5450
     97C7   C8              5451            iny
     97C8   D0 F2           5452            bne  5$          ;  do next char
                           5453
                           5454                             ;  write rest of buffer
                           5455
     97CA   B1 30           5456    7$      lda  (bufpnt),y  ;  now do buffer
     97CC   2C 180F         5457    8$      bit  pota1       ;  wait until ready
     97CF   30 FB           5458            bmi  8$
                           5459
     97D1   8D 1C01         5460            sta  data2       ;  stuff it again
                           5461
     97D4   C8              5462            iny              ;  test if done
     97D5   D0 F3           5463            bne  7$          ;  do the whole thing
                           5464
     97D7   2C 180F         5465    9$      bit  pota1       ;  wait for last char to write out
     97DA   30 FB           5466            bmi  9$
                           5467
     97DC   AD 1C0C         5468            lda  pcr2        ;  goto read mode
     97DF   09 E0           5469            ora  #$e0
     97E1   8D 1C0C         5470            sta  pcr2
                           5471
     97E4   A9 00           5472            lda  #0          ;  make data2 input $00
     97E6   8D 1C03         5473            sta  ddra2
                           5474
     97E9   20 97F9         5475            jsr  jwtobin     ;  convert write image to binary
                           5476
     97EC   A4 3F           5477            ldy  jobn        ;  make job a verify
     97EE   B9 0000         5478            lda  jobs,y
     97F1   49 30           5479            eor  #$30
     97F3   99 0000         5480            sta  jobs,y
                           5481
     97F6   4C 944F         5482            jmp  jseak       ;  scan job que
                           5483
                           5484
                           5485    ;    * jwtobin
                           5486    ;    convert write image back to
                           5487    ;    binary data
```

```
Error Addr  Code           Seq    Source statement

                          5488
                          5489
     97F9   A9 00         5490   jwtobin lda  #0
     97FB   85 2E         5491.          sta  savpnt
     97FD   85 30         5492           sta  bufpnt      ;  lsb for overflow area
     97FF   85 4F         5493           sta  nxtpnt
                          5494
     9801   A5 31         5495           lda  bufpnt+1
     9803   85 4E         5496           sta  nxtbf       ;  save for next buffer.
                          5497
     9805   A9 01         5498           lda  #>ovrbuf    ;  overflow first
     9807   85 31         5499           sta  bufpnt+1    ;  msb for overflow area
     9809   85 2F         5500           sta  savpnt+1
                          5501
     980B   A9 BB         5502           lda  #256-topwrt
     980D   85 34         5503           sta  gcrpnt      ;  offset
     980F   85 36         5504           sta  bytcnt      ;  ditto
                          5505
     9811   20 98D9       5506           jsr  jget4gb     ;  get first four- id and 3 data
                          5507
     9814   A5 52         5508           lda  btab        ;  save bid
     9816   85 38         5509           sta  bid
                          5510
     9818   A4 36         5511           ldy  bytcnt
                          5512
     981A   A5 53         5513           lda  btab+1
     981C   91 2E         5514           sta  (savpnt),y
     981E   C8            5515           iny
                          5516
     981F   A5 54         5517           lda  btab+2
     9821   91 2E         5518           sta  (savpnt),y
     9823   C8            5519           iny
                          5520
     9824   A5 55         5521           lda  btab+3
     9826   91 2E         5522           sta  (savpnt),y
     9828   C8            5523           iny
                          5524
     9829   84 36         5525           sty  bytcnt
                          5526                            .
                          5527   ; do overflow first and store back into overflow buffer
                          5528
     982B   20 98D9       5529   1$      jsr  jget4gb     ; do rest of overflow buffer
                          5530
     982E   A4 36         5531           ldy  bytcnt
                          5532
     9830   A5 52         5533           lda  btab
     9832   91 2E         5534           sta  (savpnt),y
     9834   C8            5535           iny
                          5536
     9835   A5 53         5537           lda  btab+1
```

```
Error Addr  Code          Seq   Source statement

     9837   91 2E        5538          sta  (savpnt),y
     9839   C8           5539          iny
     983A   FO OE        5540          beq  2$
                         5541
     983C   A5 54        5542          lda  btab+2
     983E   91 2E        5543          sta  (savpnt),y
     9840   C8           5544          iny
                         5545
     9841   A5 55        5546          lda  btab+3
     9843   91 2E        5547          sta  (savpnt),y
     9845   C8           5548          iny
                         5549
     9846   84 36        5550          sty  bytcnt
     9848   DO E1        5551          bne  1$         ;  jmp till end of overflow buffer
                         5552
     984A   A5 54        5553   2$     lda  btab+2
     984C   91 30        5554          sta  (bufpnt),y
     984E   C8           5555          iny
                         5556
     984F   A5 55        5557          lda  btab+3
     9851   91 30        5558          sta  (bufpnt),y
     9853   C8           5559          iny
                         5560
     9854   84 36        5561          sty  bytcnt
                         5562
     9856   20 98D9      5563   3$     jsr  jget4gb
                         5564
     9859   A4 36        5565          ldy  bytcnt
                         5566
     985B   A5 52        5567          lda  btab
     985D   91 30        5568          sta  (bufpnt),y
     985F   C8           5569          iny
                         5570
     9860   A5 53        5571          lda  btab+1
     9862   91 30        5572          sta  (bufpnt),y
     9864   C8           5573          iny
                         5574
     9865   A5 54        5575          lda  btab+2
     9867   91 30        5576          sta  (bufpnt),y
     9869   C8           5577          iny
                         5578
     986A   A5 55        5579          lda  btab+3
     986C   91 30        5580          sta  (bufpnt),y
     986E   C8           5581          iny
                         5582
     986F   84 36        5583          sty  bytcnt
     9871   CO BB        5584          cpy  #187
     9873   90 E1        5585          bcc  3$
                         5586
     9875   A9 45        5587          lda  #69              ;  move buffer up
```

```
Error Addr  Code           Seq    Source statement

     9877   85 2E         5588            sta  savpnt
                          5589
     9879   A5 31         5590            lda  bufpnt+1
     987B   85 2F         5591            sta  savpnt+1
                          5592
     987D   A0 BA         5593            ldy  #256-topwrt-1
                          5594
     987F   B1 30         5595    4$      lda  (bufpnt),y
     9881   91 2E         5596            sta  (savpnt),y
                          5597
     9883   88            5598            dey
     9884   D0 F9         5599            bne  4$
                          5600
     9886   B1 30         5601            lda  (bufpnt),y
     9888   91 2E         5602            sta  (savpnt),y
                          5603
                          5604    ; load in overflow
                          5605
     988A   A2 BB         5606            ldx  #256-topwrt
                          5607
     988C   BD 0100       5608    5$      lda  ovrbuf,x
     988F   91 30         5609            sta  (bufpnt),y
                          5610
     9891   C8            5611            iny
     9892   E8            5612            inx
     9893   D0 F7         5613            bne  5$
                          5614
     9895   86 50         5615            stx  gcrflg     ; clear buffer gcr flag
     9897   60            5616            rts
                          5617
                          5618
                          5619
                          5620    ;     * verify data block
                          5621    ;   convert to gcr verify image
                          5622    ;   test against data block
                          5623    ;   convert back to binary
                          5624
                          5625
     9898   C9 20         5626    jvrfy   cmp  #$20       ;  test if verify
     989A   F0 02         5627            beq  1$
                          5628
     989C   D0 30         5629            bne  7$         ; bra
                          5630
     989E   20 F5E9       5631    1$      jsr  chkblk     ; get block checksum
     98A1   85 3A         5632            sta  chksum
                          5633
     98A3   20 F78F       5634            jsr  bingcr     ; convert to verify image
                          5635
     98A6   20 9600       5636            jsr  jdstrt
                          5637
```

```
Error Addr  Code           Seq    Source statement

     98A9   A0 BB         5638            ldy  #256-topwrt
     98AB   B9 0100       5639    2$      lda  ovrbuf,y    ;  get char
     98AE   2C 180F       5640    3$      bit  pota1
     98B1   30 FB         5641            bmi  3$
                          5642
     98B3   4D 1C01       5643            eor  data2       ;  test if same
     98B6   D0 1C         5644            bne  4$          ; verify error
                          5645
     98B8   C8            5646            iny
     98B9   D0 F0         5647            bne  2$          ;  next byte
                          5648
                          5649
     98BB   B1 30         5650    5$      lda  (bufpnt),y ;  now do buffer
                          5651
     98BD   2C 180F       5652    6$      bit  pota1
     98C0   30 FB         5653            bmi  6$
                          5654
     98C2   4D 1C01       5655            eor  data2       ;  test if same
     98C5   D0 0D         5656            bne  4$          ;  error
                          5657
     98C7   C8            5658            iny
     98C8   C0 FD         5659            cpy  #$fd        ;  dont test off bytes
     98CA   D0 EF         5660            bne  5$
     98CC   F0 03         5661            beq  8$          ;  bra
                          5662
     98CE   20 970F       5663    7$      jsr  jsrch       ;  sector seek
     98D1   A9 01         5664    8$      lda  #1
     98D3   2C            5665            .byte  skip2
     98D4   A9 07         5666    4$      lda  #7          ;  verify error
     98D6   4C 99B5       5667            jmp  jerrr
                          5668            .include  lccgcrbn1
```

```
Error Addr  Code           Seq    Source statement

                           5670           .subttl 'lccgcrbn1.src'
                           5671
                           5672   ; even faster gcr to binary conversion
                           5673
     98D9   A4 34          5674   jget4gb ldy  gcrpnt
                           5675
     98DB   B1 30          5676           lda  (bufpnt),y
     98DD   85 56          5677           sta  gtab                ;  A
                           5678
     98DF   29 07          5679           and  #mask2
     98E1   85 57          5680           sta  gtab+1
                           5681
     98E3   C8             5682           iny                      ;  next byte
     98E4   D0 06          5683           bne  1$                  ;  test for next buffer
     98E6   A5 4E          5684           lda  nxtbf
     98E8   85 31          5685           sta  bufpnt+1
     98EA   A4 4F          5686           ldy  nxtpnt
                           5687
     98EC   B1 30          5688   1$      lda  (bufpnt),y
     98EE   85 58          5689           sta  gtab+2              ;  C
                           5690
     98F0   29 C0          5691           and  #mask2x
     98F2   05 57          5692           ora  gtab+1
     98F4   85 57          5693           sta  gtab+1              ;  B
                           5694
     98F6   A5 58          5695           lda  gtab+2
     98F8   29 01          5696           and  #mask4
     98FA   85 59          5697           sta  gtab+3
                           5698
     98FC   C8             5699           iny                      ;  next
                           5700
     98FD   B1 30          5701           lda  (bufpnt),y
     98FF   AA             5702           tax
     9900   29 F0          5703           and  #mask4x
     9902   05 59          5704           ora  gtab+3
     9904   85 59          5705           sta  gtab+3              ;  D
                           5706
     9906   8A             5707           txa
     9907   29 0F          5708           and  #mask5
     9909   85 5A          5709           sta  gtab+4
                           5710
     990B   C8             5711           iny                      ;  next byte
                           5712
     990C   B1 30          5713           lda  (bufpnt),y
     990E   85 5B          5714           sta  gtab+5              ;  F
                           5715
     9910   29 80          5716           and  #mask5x
     9912   05 5A          5717           ora  gtab+4
     9914   85 5A          5718           sta  gtab+4              ;  E
                           5719
```

```
Error Addr  Code           Seq    Source statement

     9916   A5 5B         5720           lda  gtab+5
     9918   29 03         5721           and  #mask7
     991A   85 5C         5722           sta  gtab+6
                          5723
     991C   C8            5724           iny
                          5725
                          5726   ; test for overflow during write to binary conversion
                          5727
     991D   D0 08         5728           bne  2$
                          5729
     991F   A5 4E         5730           lda  nxtbf
     9921   85 31         5731           sta  bufpnt+1
     9923   A4 4F         5732           ldy  nxtpnt
     9925   84 30         5733           sty  bufpnt
                          5734
     9927   B1 30         5735   2$      lda  (bufpnt),y
     9929   85 5D         5736           sta  gtab+7             ;  H
                          5737
     992B   29 E0         5738           and  #mask7x
     992D   05 5C         5739           ora  gtab+6
     992F   85 5C         5740           sta  gtab+6             ;  G
                          5741
     9931   C8            5742           iny
                          5743
     9932   84 34         5744           sty  gcrpnt
                          5745
     9934   A6 56         5746           ldx  gtab
     9936   BD A00D       5747           lda  gcrtb1,x           ;  a
     9939   A6 57         5748           ldx  gtab+1
     993B   1D 9F0D       5749           ora  gcrtba,x           ;  b
     993E   85 52         5750           sta  btab
                          5751
     9940   A6 58         5752           ldx  gtab+2
     9942   BD A10D       5753           lda  gcrtb2,x           ;  c
     9945   A6 59         5754           ldx  gtab+3
     9947   1D 9F0F       5755           ora  gcrtbd,x           ;  d
     994A   85 53         5756           sta  btab+1
                          5757
     994C   A6 5A         5758           ldx  gtab+4
     994E   BD 9F1D       5759           lda  gcrtbe,x           ;  e
     9951   A6 5B         5760           ldx  gtab+5
     9953   1D A20D       5761           ora  gcrtb3,x           ;  f
     9956   85 54         5762           sta  btab+2
                          5763
     9958   A6 5C         5764           ldx  gtab+6
     995A   BD 9F2A       5765           lda  gcrtbg,x           ;  g
     995D   A6 5D         5766           ldx  gtab+7
     995F   1D A30D       5767           ora  gcrtb4,x           ;  h
     9962   85 55         5768           sta  btab+3
                          5769
```

```
Error Addr  Code           Seq    Source statement

     9964   60            5770            rts
                          5771
                          5772
     9965   A9 00         5773    jgcrbin lda  #0          ;  setup pointers
     9967   85 34         5774            sta  gcrpnt
     9969   85 2E         5775            sta  savpnt
     996B   85 36         5776            sta  bytcnt
                          5777
     996D   A9 01         5778            lda  #>ovrbuf    ; point to overflow first
     996F   85 4E         5779            sta  nxtbf
                          5780
     9971   A9 BA         5781            lda  #255-toprd ; overflow offset
     9973   85 4F         5782            sta  nxtpnt
                          5783
     9975   A5 31         5784            lda  bufpnt+1       .
     9977   85 2F         5785            sta  savpnt+1
                          5786
     9979   20 98D9       5787            jsr  jget4gb
                          5788
     997C   A5 52         5789            lda  btab
     997E   85 38         5790            sta  bid         ;  get header id
                          5791
     9980   A4 36         5792            ldy  bytcnt
     9982   A5 53         5793            lda  btab+1
     9984   91 2E         5794            sta  (savpnt),y
     9986   C8            5795            iny
                          5796
     9987   A5 54         5797            lda  btab+2
     9989   91 2E         5798            sta  (savpnt),y
     998B   C8            5799            iny
                          5800
     998C   A5 55         5801            lda  btab+3
     998E   91 2E         5802            sta  (savpnt),y
     9990   C8            5803            iny
                          5804
     9991   84 36         5805    1$      sty  bytcnt
                          5806
     9993   20 98D9       5807            jsr  jget4gb
                          5808
     9996   A4 36         5809            ldy  bytcnt
                          5810
     9998   A5 52         5811            lda  btab
     999A   91 2E         5812            sta  (savpnt),y
     999C   C8            5813            iny
     999D   F0 11         5814            beq  2$                  ; test if done yet
                          5815
     999F   A5 53         5816            lda  btab+1
     99A1   91 2E         5817            sta  (savpnt),y
     99A3   C8            5818            iny
                          5819
```

```
Error Addr  Code           Seq    Source statement

     99A4   A5 54         5820            lda  btab+2
     99A6   91 2E         5821            sta  (savpnt),y
     99A8   C8            5822            iny
                          5823
     99A9   A5 55         5824            lda  btab+3
     99AB   91 2E         5825            sta  (savpnt),y
     99AD   C8            5826            iny
                          5827
     99AE   D0 E1         5828            bne  1$          ;  jmp
                          5829
     99B0   A5 2F         5830  2$        lda  savpnt+1    ; restore buffer pointer
     99B2   85 31         5831            sta  bufpnt+1
     99B4   60            5832            rts
                          5833            .include  lccutil1
```

```
Error Addr  Code           Seq    Source statement

                           5835            .subttl 'lccutil1.src'
                           5836
                           5837    ;  * utility routines
                           5838
     99B5   A4 3F          5839    jerrr   ldy  jobn       ; return  job code
     99B7   99 0000        5840            sta  jobs,y
                           5841
     99BA   A5 50          5842            lda  gcrflg     ; test if buffer left gcr
     99BC   F0 03          5843            beq  1$         ; no
                           5844
     99BE   20 97F9        5845            jsr  jwtobin    ; convert back to binary
                           5846
     99C1   20 F98F        5847    1$      jsr  trnoff     ; start timeout on drive
                           5848
     99C4   A6 49          5849            ldx  savsp
     99C6   9A             5850            txs             ; reset stack pointer
                           5851
     99C7   4C 92C8        5852            jmp  jtop       ; back to the top
                           5853
                           5854    ;   motor and stepper control
                           5855    ;   irq into controller every 8ms
                           5856
     99CA                  5857    jend
     99CA   AD 1C07        5858            lda  t1hl2      ; set irq timer
     99CD   8D 1C05        5859            sta  t1hc2
                           5860
     99D0   AD 1C00        5861            lda  dskcnt
     99D3   29 10          5862            and  #$10       ; wpsw
     99D5   C5 1E          5863            cmp  lwpt       ; same as last
     99D7   85 1E          5864            sta  lwpt       ; update
     99D9   D0 07          5865            bne  1$
                           5866
     99DB   AD 02AB        5867            lda  mtrcnt     ; anything to do?
     99DE   D0 10          5868            bne  7$         ; dec & finish up
                           5869
     99E0   F0 1C          5870            beq  2$         ; nothing to do
                           5871
     99E2   A9 FF          5872    1$      lda  #$ff
     99E4   8D 02AB        5873            sta  mtrcnt     ; 255*8ms motor on time
     99E7   20 8764        5874            jsr  moton
     99EA   A9 01          5875            lda  #1
     99EC   85 1C          5876            sta  wpsw
     99EE   D0 0E          5877            bne  2$         ; bra
                           5878
     99F0   CE 02AB        5879    7$      dec  mtrcnt     ; dec & return
     99F3   D0 09          5880            bne  2$
     99F5   A5 20          5881            lda  drvst
     99F7   C9 00          5882            cmp  #$00       ; motor off & no active drive ?
     99F9   D0 03          5883            bne  2$         ; br, do not turn it off something is going on
                           5884
```

```
Error Addr  Code           Seq    Source statement

     99FB   20 8770       5885            jsr  motoff
                          5886
     99FE   AD 02FE       5887    2$      lda  phase       ; test for phase offset
     9A01   F0 15         5888            beq  5$
                          5889
     9A03   C9 02         5890            cmp  #2
     9A05   D0 07         5891            bne  3$
                          5892
     9A07   A9 00         5893            lda  #0
     9A09   8D 02FE       5894            sta  phase
     9A0C   F0 0A         5895            beq  5$          ; bra
                          5896
     9A0E   85 4A         5897    3$      sta  steps
     9A10   A9 02         5898            lda  #2
     9A12   8D 02FE       5899            sta  phase
     9A15   4C 9A56       5900            jmp  6$
                          5901
     9A18   A6 3E         5902    5$      ldx  cdrive      ;  work on active drive only
     9A1A   30 07         5903            bmi  8$          ;  no active drive
                          5904
     9A1C   A5 20         5905            lda  drvst       ;  test if motor on
     9A1E   A8            5906            tay
     9A1F   C9 20         5907            cmp  #$20        ;  test if anything to do
     9A21   D0 03         5908            bne  9$          ;  something here
                          5909
     9A23   4C 9AC9       5910    8$      jmp  10$         ;  motor just running
                          5911
     9A26   C6 48         5912    9$      dec  acltim      ;  dec timer
     9A28   D0 1C         5913            bne  11$
                          5914
     9A2A   98            5915            tya              ;  test if acel
     9A2B   10 04         5916            bpl  12$
                          5917
                          5918
     9A2D   29 7F         5919            and  #$7f        ;  over, clear acel bit
     9A2F   85 20         5920            sta  drvst
                          5921
     9A31   29 10         5922    12$     and  #$10        ;  test if time out state
     9A33   F0 11         5923            beq  11$
                          5924
     9A35   C6 35         5925            dec  acltim2     ;  decrement second timer
     9A37   D0 0D         5926            bne  11$
                          5927
     9A39   20 8770       5928            jsr  motoff
                          5929
     9A3C   A9 FF         5930            lda  #$ff        ;  no active drive now
     9A3E   85 3E         5931            sta  cdrive
                          5932
     9A40   A9 00         5933            lda  #0          ;  drive inactive
     9A42   85 20         5934            sta  drvst       ;  clear on bit and timout
```

```
Error Addr  Code           Seq    Source statement

     9A44   F0 DD         5935            beq  8$
                         5936
     9A46   98           5937    11$     tya              ;  test if step needed
     9A47   29 40        5938            and  #$40
     9A49   D0 03        5939            bne  13$         ;  stepping
                         5940
     9A4B   4C 9AC9      5941            jmp  10$
                         5942
     9A4E   A5 62        5943    13$     lda  nxtst       ; step or settle
     9A50   D0 50        5944            bne  18$         ; go set
                         5945
     9A52   A5 4A        5946            lda  steps
     9A54   F0 43        5947            beq  17$
                         5948
     9A56   A5 4A        5949    6$      lda  steps
     9A58   10 59        5950            bpl  14$
                         5951
     9A5A   98           5952            tya
     9A5B   48           5953            pha              ; save regs .y
     9A5C   A0 63        5954            ldy  #99         ; wait for trk_00
     9A5E   AD 180F      5955    15$     lda  pota1       ; check for trk_00
     9A61   6A           5956            ror  a           ; rotate into carry
     9A62   08           5957            php              ; save it
     9A63   AD 180F      5958            lda  pota1       ; debounce it
     9A66   6A           5959            ror  a           ; => carry
     9A67   6A           5960            ror  a           ; => bit 7
     9A68   28           5961            plp              ; restore carry
     9A69   29 80        5962            and  #$80        ; set/clear sign bit
     9A6B   90 04        5963            bcc  21$
                         5964
     9A6D   10 1D        5965            bpl  16$         ; carry set(off) & sign clear(on) exit
                         5966
     9A6F   30 02        5967            bmi  20$         ; bra
                         5968
     9A71   30 19        5969    21$     bmi  16$         ; carry clear(on) & sign set(off) exit
                         5970
     9A73   88           5971    20$     dey
     9A74   D0 E8        5972            bne  15$         ; wait a while
                         5973
     9A76   B0 14        5974            bcs  16$         ; br, not track 00 ?
                         5975
     9A78   A5 7B        5976            lda  adrsed      ; enable/disable track 00 sense
     9A7A   D0 10        5977            bne  16$         ; br, nope...
                         5978
     9A7C   AD 1C00      5979            lda  dskcnt      ; phase 0
     9A7F   29 03        5980            and  #3
     9A81   D0 09        5981            bne  16$
                         5982
     9A83   68           5983            pla
     9A84   A8           5984            tay              ; restore .y
```

```
Error Addr  Code           Seq    Source statement

                          5985
      9A85   A9 00        5986            lda  #0
      9A87   85 4A        5987            sta  steps       ; nomore steps
      9A89   4C 9AC9      5988            jmp  10$
                          5989
      9A8C   68           5990    16$     pla
      9A8D   A8           5991            tay              ; restore .y
                          5992
      9A8E   E6 4A        5993            inc  steps       ; keep stepping
      9A90   AD 1C00      5994            lda  dskcnt
      9A93   38           5995            sec
      9A94   E9 01        5996            sbc  #1          ; -1 to step out
      9A96   4C 9ABB      5997            jmp  19$
                          5998
      9A99   A9 02        5999    17$     lda  #2          ;  settle time
      9A9B   85 48        6000            sta  acltim
      9A9D   85 62        6001            sta  nxtst       ; show set status
      9A9F   4C 9AC9      6002            jmp  10$
                          6003
      9AA2   C6 48        6004    18$     dec  acltim
      9AA4   D0 23        6005            bne  10$
                          6006
      9AA6   A5 20        6007            lda  drvst
      9AA8   29 BF        6008            and  #$ff-$40
      9AAA   85 20        6009            sta  drvst
                          6010
      9AAC   A9 00        6011            lda  #00
      9AAE   85 62        6012            sta  nxtst
      9AB0   4C 9AC9      6013            jmp  10$
                          6014
      9AB3   C6 4A        6015    14$     dec  steps
      9AB5   AD 1C00      6016            lda  dskcnt
      9AB8   18           6017            clc
      9AB9   69 01        6018            adc  #01
                          6019
      9ABB   29 03        6020    19$     and  #3
      9ABD   85 4B        6021            sta  tmp
      9ABF   AD 1C00      6022            lda  dskcnt
      9AC2   29 FC        6023            and  #$ff-$03    ;  mask out old
      9AC4   05 4B        6024            ora  tmp
      9AC6   8D 1C00      6025            sta  dskcnt
                          6026
      9AC9   60           6027    10$     rts
                          6028            .include  lccfmt2a
```

```
Error Addr  Code           Seq    Source statement

                           6030            .subttl 'lccfmt2a.src'
                           6031
     9ACA   21 22 23       6032   gaptbl   .byte $21,$22,$23,$24,$25,$26,$27,$28,$29
     9ACD   24 25 26
     9AD0   27 28 29
            =0009          6033   gp2cmd=*-gaptbl
     9AD3   02 02 04       6034   gp2tbl   .byte 2,2,4,6,8,8,11,19,22
     9AD6   06 08 08
     9AD9   0B 13 16
                           6035
                           6036   ; clear at least 6522 bytes written (trk1*1.08%)
                           6037
     9ADC   A0 00          6038   spdchk   ldy  #0          ; 5100/6153  write 82% sync
     9ADE   A2 1C          6039            ldx  #28
     9AE0   20 9D63        6040            jsr  jclear      ; clear whole disk & more
     9AE3   20 9B73        6041            jsr  fil_syn     ; fill with sync
     9AE6   20 FE00        6042            jsr  kill        ; go read mode
                           6043
     9AE9   A0 FF          6044            ldy  #mscnt
     9AEB   A2 FF          6045   2$       ldx  #mscnt
     9AED   2C 1C00        6046   3$       bit  dskcnt      ; try to find sync
     9AF0   10 0B          6047            bpl  5$          ; got sync
                           6048
     9AF2   CA             6049            dex
     9AF3   D0 F8          6050            bne  3$
     9AF5   88             6051            dey
     9AF6   D0 F3          6052            bne  2$
                           6053
     9AF8   A9 02          6054   4$       lda  #2
     9AFA   4C 9D59        6055            jmp  jfmte10     ; error
                           6056
     9AFD   A0 00          6057   5$       ldy  #0
     9AFF   A2 00          6058            ldx  #0
     9B01   2C 1C00        6059   6$       bit  dskcnt      ; wait for sync to go away
     9B04   10 FB          6060            bpl  6$          ; got sync
                           6061
     9B06   AD 1C00        6062   7$       lda  dskcnt      ; count time till next sync
     9B09   C8             6063            iny              ; lsb not used
     9B0A   D0 01          6064            bne  8$
     9B0C   E8             6065            inx              ; msb
     9B0D   29 80          6066   8$       and  #$80
     9B0F   D0 F5          6067            bne  7$          ; br, still no sync
                           6068
     9B11   A9 00          6069            lda  #0
     9B13   85 3B          6070            sta  tsttrk
     9B15   8A             6071            txa              ; msb
                           6072
                           6073   ;*********************************************************
                           6074   ;*           ---  lookup gap2 in table ---              *
                           6075   ;*       x :  21 / 22 / 23 / 24 / 25 / 26 / 27 / 28 / 29 *
```

```
Error Addr  Code           Seq    Source statement

                           6076   ;*  gap2   :  02 / 02 / 04 / 06 / 08 / 08 / 0b / 13 / 16 *
                           6077   ;*  speed  : fast ------------------------------ slow  *
                           6078   ;*********************************************************
     9B16   A2 08          6079           ldx  #gp2cmd-1
     9B18   DD 9ACA        6080   9$      cmp  gaptbl,x    ; lookup timing gap
     9B1B   F0 05          6081           beq  10$
     9B1D   CA             6082           dex
     9B1E   10 F8          6083           bpl  9$
     9B20   30 D6          6084           bmi  4$          ; none
                           6085
     9B22   BD 9AD3        6086   10$     lda  gp2tbl,x    ; look it up
     9B25   8D 0626        6087           sta  dtrck
     9B28   60             6088           rts
                           6089
     9B29   A5 51          6090   jformt  lda  ftnum       ; test if formating done
     9B2B   10 2B          6091           bpl  1$          ; yes
                           6092
     9B2D   A9 60          6093           lda  #$60        ; status = stepping
     9B2F   85 20          6094           sta  drvst
                           6095
     9B31   AD 01B2        6096           lda  fmtsid
     9B34   D0 03          6097           bne  3$
                           6098
     9B36   A9 01          6099           lda  #01
     9B38   2C             6100           .byte  skip2
     9B39   A9 24          6101   3$      lda  #36
     9B3B   85 22          6102           sta  drvtrk      ; track 1/36
     9B3D   85 51          6103           sta  ftnum       ; track count =1/36
     9B3F   C9 24          6104           cmp  #36         ; set/clr carry flag
     9B41   20 93F3        6105           jsr  set_side
                           6106
     9B44   A9 A4          6107           lda  #256-92     ; step back 45 tracks
     9B46   85 4A          6108           sta  steps
                           6109
     9B48   AD 1C00        6110           lda  dskcnt      ; phase a
     9B4B   29 FC          6111           and  #$ff-03
     9B4D   8D 1C00        6112           sta  dskcnt
                           6113
     9B50   A9 0A          6114           lda  #10
     9B52   8D 0620        6115           sta  cnt         ; init error count
     9B55   4C 99CA        6116           jmp  jend        ; go back to controler
                           6117
     9B58   A0 00          6118   1$      ldy  #00
     9B5A   B1 32          6119           lda  (hdrpnt),y
     9B5C   C5 51          6120           cmp  ftnum
     9B5E   F0 07          6121           beq  2$
                           6122
     9B60   A5 51          6123           lda  ftnum
     9B62   91 32          6124           sta  (hdrpnt),y
     9B64   4C 99CA        6125           jmp  jend
```

```
Error Addr  Code           Seq    Source statement

                           6126
      9B67   AD 1C00       6127   2$       lda  dskcnt      ; check wps
      9B6A   29 10         6128            and  #$10
      9B6C   D0 1B         6129            bne  toptst     ; ok
                           6130
      9B6E   A9 08         6131            lda  #8
      9B70   4C 9D51       6132            jmp  jfmterr    ; wp error
                           6133
      9B73   A2 14         6134   fil_syn  ldx  #20        ; 20*256 bytes of sync
      9B75   A9 FF         6135            lda  #$ff       ; write syncs
      9B77   2C 180F       6136   1$       bit  pota1
      9B7A   30 FB         6137            bmi  1$
                           6138
      9B7C   8D 1C01       6139            sta  data2
      9B7F   2C 1C00       6140            bit  byt_clr
                           6141
      9B82   88            6142            dey
      9B83   D0 F2         6143            bne  1$
      9B85   CA            6144            dex
      9B86   D0 EF         6145            bne  1$
      9B88   60            6146            rts
                           6147
      9B89   A5 3B         6148   toptst   lda  tsttrk
      9B8B   10 03         6149            bpl  1$
                           6150
      9B8D   20 9ADC       6151            jsr  spdchk
                           6152
      9B90   AD 0626       6153   1$       lda  dtrck
                           6154                         ;  create header images
      9B93   18            6155            clc
      9B94   A9 03         6156            lda  #>bufs
      9B96   85 33         6157            sta  hdrpnt+1
      9B98   A9 00         6158            lda  #00
      9B9A   85 32         6159            sta  hdrpnt     ; point hdr point to buf0
      9B9C   8D 0628       6160            sta  sect
                           6161
      9B9F   A0 00         6162            ldy  #0
      9BA1   A5 39         6163   2$       lda  hbid       ; hbid cs s t id id 0f 0f
      9BA3   91 32         6164            sta  (hdrpnt),y
      9BA5   C8            6165            iny
      9BA6   A9 00         6166            lda  #00        ; check sum is zero for now
      9BA8   91 32         6167            sta  (hdrpnt),y
      9BAA   C8            6168            iny
                           6169
      9BAB   AD 0628       6170            lda  sect       ; store sector #
      9BAE   91 32         6171            sta  (hdrpnt),y
      9BB0   C8            6172            iny
                           6173
      9BB1   A5 51         6174            lda  ftnum      ; store track #
      9BB3   91 32         6175            sta  (hdrpnt),y
```

```
Error Addr  Code           Seq    Source statement

     9BB5   C8            6176            iny
                          6177
     9BB6   A5 13         6178            lda  dskid+1    ; store id low
     9BB8   91 32         6179            sta  (hdrpnt),y
     9BBA   C8            6180            iny
                          6181
     9BBB   A5 12         6182            lda  dskid      ; store id hi
     9BBD   91 32         6183            sta  (hdrpnt),y
     9BBF   C8            6184            iny
                          6185
     9BC0   A9 0F         6186            lda  #$0f       ; store gap1 bytes
     9BC2   91 32         6187            sta  (hdrpnt),y
     9BC4   C8            6188            iny
     9BC5   91 32         6189            sta  (hdrpnt),y
     9BC7   C8            6190            iny
                          6191
     9BC8   98            6192            tya             ; save this point
     9BC9   48            6193            pha
     9BCA   A2 07         6194            ldx  #07
     9BCC   A9 00         6195            lda  #00
     9BCE   85 3A         6196            sta  chksum     ; zero checksum
     9BD0   88            6197   3$       dey
     9BD1   B1 32         6198            lda  (hdrpnt),y
     9BD3   45 3A         6199            eor  chksum
     9BD5   85 3A         6200            sta  chksum
     9BD7   CA            6201            dex
     9BD8   D0 F6         6202            bne  3$
                          6203
     9BDA   91 32         6204            sta  (hdrpnt),y ; store checksum
     9BDC   68            6205            pla
     9BDD   A8            6206            tay             ; restore pointer
                          6207
     9BDE   EE 0628       6208            inc  sect       ; goto next sector
                          6209
     9BE1   AD 0628       6210            lda  sect       ; test if done yet
     9BE4   C5 43         6211            cmp  sectr
     9BE6   90 B9         6212            bcc  2$         ; more to do
                          6213
     9BE8   A9 03         6214            lda  #3
     9BEA   85 31         6215            sta  bufpnt+1   ; $03XX
                          6216
     9BEC   20 FE30       6217            jsr  fbtog      ; convert to gcr with no bid char
                          6218                            ; move buffer up 79 bytes
                          6219
     9BEF   A0 BA         6220            ldy  #$ff-69    ; for i=n-1 to 0:mem i+69 :=mem i :next
     9BF1   B1 32         6221   4$       lda  (hdrpnt),y ; move buf0 up 69 bytes
     9BF3   A2 45         6222            ldx  #69
     9BF5   86 32         6223            stx  hdrpnt
     9BF7   91 32         6224            sta  (hdrpnt),y
     9BF9   A2 00         6225            ldx  #00
```

```
Error Addr  Code           Seq    Source statement

     9BFB   86 32         6226            stx  hdrpnt
     9BFD   88             6227            dey
     9BFE   C0 FF          6228            cpy  #$ff
     9C00   D0 EF          6229            bne  4$
                           6230
     9C02   A0 44          6231            ldy  #68        ; #bytes to move
     9C04   B9 01BB        6232   5$       lda  ovrbuf+$bb,y
     9C07   91 32          6233            sta  (hdrpnt),y
     9C09   88             6234            dey
     9C0A   10 F8          6235            bpl  5$
                           6236
                           6237   ;   create data block of zero
                           6238
     9C0C   18             6239            clc
     9C0D   A9 03          6240            lda  #>bufs
     9C0F   69 02          6241            adc  #02
     9C11   85 31          6242            sta  bufpnt+1   ; point to buf2
     9C13   A9 00          6243            lda  #00
     9C15   A8             6244            tay             ; init offset
     9C16   91 30          6245   6$       sta  (bufpnt),y
     9C18   C8             6246            iny
     9C19   D0 FB          6247            bne  6$
                           6248
                           6249
                           6250   ;   convert data block to gcr
                           6251   ;   write image
                           6252   ;   leave it in ovrbuf and buffer
                           6253
     9C1B   20 F5E9        6254            jsr  chkblk     ; get block checksum
     9C1E   85 3A          6255            sta  chksum
     9C20   20 F78F        6256            jsr  bingcr
                           6257
                           6258   ;   start the format now
                           6259   ;   write out sync header gap1
                           6260   ;   data block
                           6261
     9C23   A9 00          6262            lda  #0         ; init counter
     9C25   85 1B          6263            sta  fmhdpt
                           6264
     9C27   A2 06          6265            ldx  #6         ; clear 8% of disk
     9C29   20 9D63        6266            jsr  jclear     ; clear disk
                           6267
     9C2C   A0 05          6268   7$       ldy  #numsyn    ; write 4 gcr bytes
                           6269
     9C2E   2C 180F        6270   8$       bit  pota1
     9C31   30 FB          6271            bmi  8$
                           6272
     9C33   A9 FF          6273            lda  #$ff       ; write sync
     9C35   8D 1C01        6274            sta  data2
     9C38   2C 1C00        6275            bit  byt_clr    ; clear pa latch
```

```
Error Addr  Code           Seq    Source statement

                           6276
      9C3B  88             6277            dey
      9C3C  DO FO          6278            bne  8$
                           6279
      9C3E  A2 OA          6280            ldx  #10        ; write out header
      9C4O  A4 1B          6281            ldy  fmhdpt
                           6282
      9C42  2C 18OF        6283   9$       bit  pota1
      9C45  3O FB          6284            bmi  9$
                           6285
      9C47  B1 32          6286            lda  (hdrpnt),y ; get header data
      9C49  8D 1CO1        6287            sta  data2
      9C4C  2C 1COO        6288            bit  byt_clr
      9C4F  C8             6289            iny
                           6290
      9C5O  CA             6291            dex
      9C51  DO EF          6292            bne  9$
                           6293
                           6294
                           6295   ; * write out gap1
                           6296
      9C53  AO 09          6297            ldy  #gap1-2    ; write  gcr bytes
                           6298
      9C55  2C 18OF        6299   10$      bit  pota1
      9C58  3O FB          6300            bmi  10$
                           6301
      9C5A  A9 55          6302            lda  #$55
      9C5C  8D 1CO1        6303            sta  data2
      9C5F  2C 1COO        6304            bit  byt_clr
                           6305
      9C62  88             6306            dey
      9C63  DO FO          6307            bne  10$
                           6308
                           6309   ; * write out data block
                           6310
      9C65  A9 FF          6311            lda  #$ff       ; write data block sync
      9C67  AO O5          6312            ldy  #numsyn
                           6313
      9C69  2C 18OF        6314   11$      bit  pota1
      9C6C  3O FB          6315            bmi  11$
                           6316
      9C6E  8D 1CO1        6317            sta  data2
      9C71  2C 1COO        6318            bit  byt_clr
                           6319
      9C74  88             6320            dey
      9C75  DO F2          6321            bne  11$
                           6322
      9C77  AO BB          6323            ldy  #$bb       ; write out ovrbuf
      9C79  2C 18OF        6324   12$      bit  pota1
      9C7C  3O FB          6325            bmi  12$
```

```
Error Addr  Code           Seq    Source statement

                           6326
      9C7E   B9 0100       6327            lda  ovrbuf,y
      9C81   8D 1C01       6328            sta  data2
      9C84   2C 1C00       6329            bit  byt_clr
                           6330
      9C87   C8            6331            iny
      9C88   D0 EF         6332            bne  12$
                           6333
      9C8A   2C 180F       6334   13$      bit  pota1
      9C8D   30 FB         6335            bmi  13$
                           6336
      9C8F   B1 30         6337            lda  (bufpnt),y
      9C91   8D 1C01       6338            sta  data2
      9C94   2C 1C00       6339            bit  byt_clr
                           6340
      9C97   C8            6341            iny
      9C98   D0 F0         6342            bne  13$
                           6343
      9C9A   A9 55         6344            lda  #$55       ; write gap2(dtrck)
      9C9C   AC 0626       6345            ldy  dtrck
                           6346
      9C9F   2C 180F       6347   14$      bit  pota1
      9CA2   30 FB         6348            bmi  14$
                           6349
      9CA4   8D 1C01       6350            sta  data2
      9CA7   2C 1C00       6351            bit  byt_clr
                           6352
      9CAA   88            6353            dey
      9CAB   D0 F2         6354            bne  14$
                           6355
                           6356
      9CAD   A5 1B         6357            lda  fmhdpt     ; advance header pointer
      9CAF   18            6358            clc
      9CB0   69 0A         6359            adc  #10
      9CB2   85 1B         6360            sta  fmhdpt
                           6361
                           6362   ;  done writing sector
                           6363
      9CB4   CE 0628       6364            dec  sect       ; go to next on
      9CB7   F0 03         6365            beq  15$        ; br, no more to do
                           6366
      9CB9   4C 9C2C       6367            jmp  7$
                           6368
      9CBC   2C 180F       6369   15$      bit  pota1      ; wait for last one to write
      9CBF   30 FB         6370            bmi  15$
                           6371
      9CC1   2C 1C00       6372            bit  byt_clr
                           6373
      9CC4   2C 180F       6374   16$      bit  pota1      ;wait for last one to write
      9CC7   30 FB         6375            bmi  16$
```

```
Error Addr  Code           Seq    Source statement

                          6376
     9CC9   2C 1C00       6377            bit  byt_clr
                          6378
     9CCC   20 FE00       6379            jsr  kill       ; goto read mode
                          6380            .include  lccfmt3a
```

```
Error Addr  Code           Seq    Source statement

                          6382            .subttl 'lccfmt3a.src'
                          6383
                          6384    ;*   format done, now verify it
                          6385
     9CCF   A9 C8         6386    vfmt    lda  #200       ;  look at 200 syncs
     9CD1   8D 0623       6387            sta  trys
                          6388
     9CD4   A9 00         6389    1$      lda  #00
     9CD6   85 1B         6390            sta  fmhdpt     ; start with first sector again
     9CD8   A5 43         6391            lda  sectr      ;  sector counter
     9CDA   8D 0628       6392            sta  sect
     9CDD   20 9754       6393    2$      jsr  jsync      ;  find sync
     9CE0   A2 0A         6394            ldx  #10
     9CE2   A4 1B         6395            ldy  fmhdpt     ; current header pointer
                          6396
     9CE4   B1 32         6397    3$      lda  (hdrpnt),y
     9CE6   2C 180F       6398    4$      bit  pota1      ;  get header byte
     9CE9   30 FB         6399            bmi  4$
                          6400
     9CEB   CD 1C01       6401            cmp  data2
     9CEE   D0 0E         6402            bne  5$         ; error
                          6403
     9CF0   C8            6404            iny
     9CF1   CA            6405            dex
     9CF2   D0 F0         6406            bne  3$         ;  test all bytes
                          6407
     9CF4   18            6408            clc             ; update headr pointer
     9CF5   A5 1B         6409            lda  fmhdpt
     9CF7   69 0A         6410            adc  #10
     9CF9   85 1B         6411            sta  fmhdpt
                          6412
     9CFB   4C 9D08       6413            jmp  6$         ;  now test data
                          6414
     9CFE   CE 0623       6415    5$      dec  trys       ;  test if too many errors
     9D01   D0 D1         6416            bne  1$
                          6417
     9D03   A9 06         6418            lda  #notfnd    ;  too many error
     9D05   4C 9D51       6419            jmp  jfmterr
                          6420
     9D08   20 9754       6421    6$      jsr  jsync      ;  find data sync
                          6422
     9D0B   A0 BB         6423            ldy  #256-topwrt
     9D0D   B9 0100       6424    7$      lda  ovrbuf,y   ; ovr buff offset
     9D10   2C 180F       6425    8$      bit  pota1
     9D13   30 FB         6426            bmi  8$
                          6427
     9D15   CD 1C01       6428            cmp  data2      ;  compare gcr
     9D18   D0 E4         6429            bne  5$         ; error
                          6430
     9D1A   C8            6431            iny
```

```
Error Addr  Code           Seq    Source statement

     9D1B   D0 F0         6432            bne  7$          ;  do all ovrbuf
                          6433
     9D1D   B1 30         6434    9$      lda  (bufpnt),y
     9D1F   2C 180F       6435    10$     bit  pota1
     9D22   30 FB         6436            bmi  10$
                          6437
     9D24   CD 1C01       6438            cmp  data2
     9D27   D0 D5         6439            bne  5$
                          6440
     9D29   C8            6441            iny
     9D2A   D0 F1         6442            bne  9$
                          6443
     9D2C   CE 0628       6444            dec  sect       ; more sectors to test?
     9D2F   D0 AC         6445            bne  2$         ; yes
                          6446
                          6447    ;  all sectors done ok
                          6448
     9D31   E6 51         6449            inc  ftnum      ;  goto next track
     9D33   A5 51         6450            lda  ftnum
     9D35   2C 01B1       6451            bit  side       ;  what side are we on ?
     9D38   30 03         6452            bmi  13$
                          6453
     9D3A   C9 24         6454            cmp  #36        ;  #tracks max
     9D3C   2C            6455            .byte skip2
     9D3D   C9 47         6456    13$     cmp  #71
     9D3F   B0 03         6457            bcs  12$
                          6458
     9D41   4C 99CA       6459            jmp  jend       ;  more to do
                          6460
                          6461
     9D44   A9 FF         6462    12$     lda  #$ff       ;  clear ftnum
     9D46   85 51         6463            sta  ftnum
                          6464
     9D48   A9 00         6465            lda  #$0        ;  clear gcr buffer flag
     9D4A   85 50         6466            sta  gcrflg
                          6467
     9D4C   A9 01         6468            lda  #1         ;  return ok code
     9D4E   4C 99B5       6469            jmp  jerrr
                          6470            .include  lccfmt4a
```

```
Error Addr  Code           Seq    Source statement

                          6472            .subttl 'lccfmt4a.src'
                          6473
     9D51   CE 0620       6474    jfmterr dec  cnt         ;  test for retry
     9D54   F0 03         6475            beq  jfmte10
                          6476
     9D56   4C 99CA       6477            jmp  jend
                          6478
     9D59   A0 FF         6479    jfmte10 ldy  #$ff
     9D5B   84 51         6480            sty  ftnum       ;  clear format
                          6481
     9D5D   C8            6482            iny
     9D5E   84 50         6483            sty  gcrflg
                          6484
     9D60   4C 99B5       6485            jmp  jerrr
                          6486
                          6487    ; this subroutine will be called with x*255 for amount of chars written
                          6488
     9D63   AD 1C0C       6489    jclear  lda  pcr2        ;  enable write
     9D66   29 1F         6490            and  #$ff-$e0    ;  wr mode=0
     9D68   09 C0         6491            ora  #$c0
     9D6A   8D 1C0C       6492            sta  pcr2
                          6493
     9D6D   A9 FF         6494            lda  #$ff        ;  make port an output
     9D6F   8D 1C03       6495            sta  ddra2       ;  clear pending
                          6496
     9D72   A9 55         6497            lda  #$55        ;  write a 1f pattern
     9D74   A0 00         6498            ldy  #00
     9D76   2C 180F       6499    1$      bit  pota1
     9D79   30 FB         6500            bmi  1$
                          6501
     9D7B   2C 1C00       6502            bit  byt_clr
     9D7E   8D 1C01       6503            sta  data2
                          6504
     9D81   88            6505            dey
     9D82   D0 F2         6506            bne  1$
                          6507
     9D84   CA            6508            dex              ;  dex amount * 255
     9D85   D0 EF         6509            bne  1$
                          6510
     9D87   60            6511            rts
                          6512            .include  irq1541
```

```
Error Addr  Code           Seq    Source statement

                           6514           .subttl 'irq1541.src'
      9D88  48             6515   irq     pha
      9D89  8A             6516           txa
      9D8A  48             6517           pha
      9D8B  98             6518           tya
      9D8C  48             6519           pha
                           6520
      9D8D  AD 400D        6521           lda  icr
      9D90  29 08          6522           and  #8
      9D92  F0 26          6523           beq  1$          ; fast serial request
                           6524
      9D94  2C 02AF        6525           bit  lock        ; locked ?
      9D97  30 21          6526           bmi  1$
                           6527
      9D99  AD 180F        6528           lda  pota1       ; change to 2 Mhz
      9D9C  09 20          6529           ora  #$20
      9D9E  8D 180F        6530           sta  pota1
      9DA1  A9 DE          6531           lda  #<jirq
      9DA3  8D 02A9        6532           sta  irqjmp
      9DA6  A9 9D          6533           lda  #>jirq
      9DA8  8D 02AA        6534           sta  irqjmp+1    ; re-vector irq
                           6535
      9DAB  A9 40          6536           lda  #tim2       ; 8 ms irq's at 2 Mhz - controller irq's
      9DAD  8D 1C07        6537           sta  tihl2
      9DB0  8D 1C05        6538           sta  tihc2       ; adjust timers for 2 Mhz
      9DB3  A9 00          6539           lda  #0
      9DB5  85 62          6540           sta  nxtst       ; not a vector
      9DB7  4C 9DEA        6541           jmp  irq_0
                           6542
      9DBA  AD 180D        6543   1$      lda  ifr1
      9DBD  29 02          6544           and  #2
      9DBF  F0 03          6545           beq  2$          ;  not atn
                           6546
      9DC1  20 E853        6547           jsr  atnirq      ;  handle atn request
                           6548
      9DC4  AD 1C0D        6549   2$      lda  ifr2        ;  test if timer
      9DC7  0A             6550           asl  a
      9DC8  10 03          6551           bpl  3$          ;  not timer
                           6552
      9DCA  20 F2B0        6553           jsr  lcc         ;  goto controller
                           6554
      9DCD  BA             6555   3$      tsx
      9DCE  BD 0104        6556           lda  $0104,x     ; check processor break flag
      9DD1  29 10          6557           and  #$10
      9DD3  F0 03          6558           beq  4$
                           6559
      9DD5  20 F2B0        6560           jsr  lcc
                           6561
      9DD8  68             6562   4$      pla              ; restore .y, .x, .a
      9DD9  A8             6563           tay
```

```
Error Addr  Code          Seq    Source statement

     9DDA   68            6564           pla
     9DDB   AA            6565           tax
     9DDC   68            6566           pla
     9DDD   40            6567           rti
                          6568           .include  irq1571
```

```
Error Addr  Code           Seq    Source statement

                           6570           .subttl 'irq1571.src'
                           6571
     9DDE   48             6572   jirq    pha             ; save .a,.x,.y
     9DDF   8A             6573           txa
     9DE0   48             6574           pha
     9DE1   98             6575           tya
     9DE2   48             6576           pha
                           6577
     9DE3   AD 400D        6578           lda  icr        ; chk for fast byte
     9DE6   29 08          6579           and  #8
     9DE8   F0 08          6580           beq  irq_1
                           6581
     9DEA   A5 37          6582   irq_0   lda  fastsr
     9DEC   09 40          6583           ora  #$40       ; set byte in flag
     9DEE   85 37          6584           sta  fastsr
     9DF0   D0 22          6585           bne  irq_2      ; bra
                           6586
     9DF2   AD 180D        6587   irq_1   lda  ifr1       ; test for atn
     9DF5   29 02          6588           and  #2
     9DF7   F0 07          6589           beq  1$
                           6590
     9DF9   2C 1801        6591           bit  pa1        ; clear (ca1)
     9DFC   A9 01          6592           lda  #1         ; handle atn request
     9DFE   85 7C          6593           sta  atnpnd     ; flag atn is pending
                           6594
     9E00   BA             6595   1$      tsx             ; check break flag
     9E01   BD 0104        6596           lda  $0104,x    ; check processor status
     9E04   29 10          6597           and  #$10
     9E06   F0 03          6598           beq  2$
                           6599
     9E08   20 92BA        6600           jsr  jlcc       ; forced irq do controller job
                           6601
     9E0B   AD 1C0D        6602   2$      lda  ifr2       ; test if timer
     9E0E   0A             6603           asl  a
     9E0F   10 03          6604           bpl  3$
                           6605
     9E11   20 92BA        6606           jsr  jlcc       ; goto controller
                           6607
     9E14                  6608   3$
     9E14   68             6609   irq_2   pla             ; restore .y,.x,.a
     9E15   A8             6610           tay
     9E16   68             6611           pla
     9E17   AA             6612           tax
     9E18   68             6613           pla
     9E19   40             6614           rti
                           6615           .include  gcrtables
```

```
Error Addr  Code           Seq    Source statement

                           6617           .subttl 'gcr.tables'
                           6618
            =9F00          6619   *=*+256-<*              ; even page boundary
                           6620
      9F00  FF             6621   gcrtbh  .byte    $ff     ; -d
      9F01  FF             6622           .byte    $ff     ; -c
      9F02  FF             6623           .byte    $ff     ; -b
      9F03  FF             6624           .byte    $ff     ; -a
      9F04  FF             6625           .byte    $ff     ; -9
      9F05  FF             6626           .byte    $ff     ; -8
      9F06  FF             6627           .byte    $ff     ; -7
      9F07  FF             6628           .byte    $ff     ; -6
      9F08  FF             6629           .byte    $ff     ; -5
      9F09  08             6630           .byte    $08     ; -4  h
      9F0A  00             6631           .byte    $00     ; -3  h
      9F0B  01             6632           .byte    $01     ; -2  h
      9F0C  FF             6633           .byte    $ff     ; -1
      9F0D  0C             6634   gcrtba  .byte    $0c     ; 0  h
      9F0E  04             6635   gcrtbf  .byte    $04     ; 1  h
      9F0F  05             6636   gcrtbd  .byte    $05     ; 2  h
      9F10  FF             6637           .byte    $ff     ; 4
      9F11  FF             6638           .byte    $ff     ; 3
      9F12  02             6639           .byte    $02     ; 5  h
      9F13  03             6640           .byte    $03     ; 6  h
      9F14  FF             6641           .byte    $ff     ; 7
      9F15  0F             6642           .byte    $0f     ; 8  h
      9F16  06             6643           .byte    $06     ; 9  h
      9F17  07             6644           .byte    $07     ; a  h
      9F18  FF             6645           .byte    $ff     ; b
      9F19  09             6646           .byte    $09     ; c  h
      9F1A  0A             6647           .byte    $0a     ; d  h
      9F1B  0B             6648           .byte    $0b     ; e  h
      9F1C  FF             6649           .byte    $ff     ; f
      9F1D  0D             6650   gcrtbe  .byte    $0d     ; 10  h
      9F1E  0E             6651           .byte    $0e     ; 11  h
      9F1F  80             6652           .byte    $80     ; 12  c
      9F20  FF             6653           .byte    $ff     ; 13
      9F21  00             6654           .byte    $00     ; 14  c
      9F22  00             6655           .byte    $00     ; 15  e
      9F23  10             6656           .byte    $10     ; 16  c
      9F24  40             6657           .byte    $40     ; 17  e
      9F25  FF             6658           .byte    $ff     ; 18
      9F26  20             6659           .byte    $20     ; 19  e
      9F27  C0             6660           .byte    $c0     ; 1a  c
      9F28  60             6661           .byte    $60     ; 1b  e
      9F29  40             6662           .byte    $40     ; 1c  c
      9F2A  A0             6663   gcrtbg  .byte    $a0     ; 1d  e
      9F2B  50             6664           .byte    $50     ; 1e  c
      9F2C  E0             6665           .byte    $e0     ; 1f  e
      9F2D  FF             6666           .byte    $ff     ; 20
```

```
Error Addr  Code          Seq    Source statement

     9F2E   FF          6667            .byte   $ff     ; 21
     9F2F   FF          6668            .byte   $ff     ; 22
     9F30   02          6669            .byte   $02     ; 23  d
     9F31   20          6670            .byte   $20     ; 24  c
     9F32   08          6671            .byte   $08     ; 25  f
     9F33   30          6672            .byte   $30     ; 26  c
     9F34   FF          6673            .byte   $ff     ; 27
     9F35   FF          6674            .byte   $ff     ; 28
     9F36   00          6675            .byte   $00     ; 29  f
     9F37   F0          6676            .byte   $f0     ; 2a  c
     9F38   FF          6677            .byte   $ff     ; 2b
     9F39   60          6678            .byte   $60     ; 2c  c
     9F3A   01          6679            .byte   $01     ; 2d  f
     9F3B   70          6680            .byte   $70     ; 2e  c
     9F3C   FF          6681            .byte   $ff     ; 2f
     9F3D   FF          6682            .byte   $ff     ; 30
     9F3E   FF          6683            .byte   $ff     ; 31
     9F3F   90          6684            .byte   $90     ; 32  c
     9F40   03          6685            .byte   $03     ; 33  d
     9F41   A0          6686            .byte   $a0     ; 34  c
     9F42   0C          6687            .byte   $0c     ; 35  f
     9F43   B0          6688            .byte   $b0     ; 36  c
     9F44   FF          6689            .byte   $ff     ; 37
     9F45   FF          6690            .byte   $ff     ; 38
     9F46   04          6691            .byte   $04     ; 39  f
     9F47   D0          6692            .byte   $d0     ; 3a  c
     9F48   FF          6693            .byte   $ff     ; 3b
     9F49   E0          6694            .byte   $e0     ; 3c  c
     9F4A   05          6695            .byte   $05     ; 3d  f
     9F4B   80          6696            .byte   $80     ; 3e  g
     9F4C   FF          6697            .byte   $ff     ; 3f
     9F4D   90          6698            .byte   $90     ; 40  g
     9F4E   FF          6699            .byte   $ff     ; 41
     9F4F   08          6700            .byte   $08     ; 42  b
     9F50   0C          6701            .byte   $0c     ; 43  b
     9F51   FF          6702            .byte   $ff     ; 44
     9F52   0F          6703            .byte   $0f     ; 45  b
     9F53   09          6704            .byte   $09     ; 46  b
     9F54   0D          6705            .byte   $0d     ; 47  b
     9F55   80          6706            .byte   $80     ; 48  a
     9F56   02          6707            .byte   $02     ; 49  f
     9F57   FF          6708            .byte   $ff     ; 4a
     9F58   FF          6709            .byte   $ff     ; 4b
     9F59   FF          6710            .byte   $ff     ; 4c
     9F5A   03          6711            .byte   $03     ; 4d  f
     9F5B   FF          6712            .byte   $ff     ; 4e
     9F5C   FF          6713            .byte   $ff     ; 4f
     9F5D   00          6714            .byte   $00     ; 50  a
     9F5E   FF          6715            .byte   $ff     ; 51
     9F5F   FF          6716            .byte   $ff     ; 52
```

```
Error Addr  Code           Seq    Source statement

      9F60  OF            6717            .byte   $0f     ; 53  d
      9F61  FF            6718            .byte   $ff     ; 54
      9F62  OF            6719            .byte   $0f     ; 55  f
      9F63  FF            6720            .byte   $ff     ; 56
      9F64  FF            6721            .byte   $ff     ; 57
      9F65  10            6722            .byte   $10     ; 58  a
      9F66  06            6723            .byte   $06     ; 59  f
      9F67  FF            6724            .byte   $ff     ; 5a
      9F68  FF            6725            .byte   $ff     ; 5b
      9F69  FF            6726            .byte   $ff     ; 5c
      9F6A  07            6727            .byte   $07     ; 5d  f
      9F6B  00            6728            .byte   $00     ; 5e  g
      9F6C  20            6729            .byte   $20     ; 5f  g
      9F6D  A0            6730            .byte   $a0     ; 60  g
      9F6E  FF            6731            .byte   $ff     ; 61
      9F6F  FF            6732            .byte   $ff     ; 62
      9F70  06            6733            .byte   $06     ; 63  d
      9F71  FF            6734            .byte   $ff     ; 64
      9F72  09            6735            .byte   $09     ; 65  f
      9F73  FF            6736            .byte   $ff     ; 66
      9F74  FF            6737            .byte   $ff     ; 67
      9F75  C0            6738            .byte   $c0     ; 68  a
      9F76  0A            6739            .byte   $0a     ; 69  f
      9F77  FF            6740            .byte   $ff     ; 6a
      9F78  FF            6741            .byte   $ff     ; 6b
      9F79  FF            6742            .byte   $ff     ; 6c
      9F7A  0B            6743            .byte   $0b     ; 6d  f
      9F7B  FF            6744            .byte   $ff     ; 6e
      9F7C  FF            6745            .byte   $ff     ; 6f
      9F7D  40            6746            .byte   $40     ; 70  a
      9F7E  FF            6747            .byte   $ff     ; 71
      9F7F  FF            6748            .byte   $ff     ; 72
      9F80  07            6749            .byte   $07     ; 73  d
      9F81  FF            6750            .byte   $ff     ; 74
      9F82  0D            6751            .byte   $0d     ; 75  f
      9F83  FF            6752            .byte   $ff     ; 76
      9F84  FF            6753            .byte   $ff     ; 77
      9F85  50            6754            .byte   $50     ; 78  a
      9F86  0E            6755            .byte   $0e     ; 79  f
      9F87  FF            6756            .byte   $ff     ; 7a
      9F88  FF            6757            .byte   $ff     ; 7b
      9F89  FF            6758            .byte   $ff     ; 7c
      9F8A  FF            6759            .byte   $ff     ; 7d
      9F8B  10            6760            .byte   $10     ; 7e  g
      9F8C  30            6761            .byte   $30     ; 7f  g
      9F8D  B0            6762            .byte   $b0     ; 80  g
      9F8E  FF            6763            .byte   $ff     ; 81
      9F8F  00            6764            .byte   $00     ; 82  b
      9F90  04            6765            .byte   $04     ; 83  b
      9F91  02            6766            .byte   $02     ; 84  b
```

```
Error Addr  Code           Seq    Source statement

     9F92   06             6767            .byte    $06    ; 85  b
     9F93   0A             6768            .byte    $0a    ; 86  b
     9F94   0E             6769            .byte    $0e    ; 87  b
     9F95   80             6770            .byte    $80    ; 88  a
     9F96   FF             6771            .byte    $ff    ; 89
     9F97   FF             6772            .byte    $ff    ; 8a
     9F98   FF             6773            .byte    $ff    ; 8b
     9F99   FF             6774            .byte    $ff    ; 8c
     9F9A   FF             6775            .byte    $ff    ; 8d
     9F9B   FF             6776            .byte    $ff    ; 8e
     9F9C   FF             6777            .byte    $ff    ; 8f
     9F9D   20             6778            .byte    $20    ; 90  a
     9F9E   FF             6779            .byte    $ff    ; 91
     9F9F   08             6780            .byte    $08    ; 92  d
     9FA0   09             6781            .byte    $09    ; 93  d
     9FA1   80             6782            .byte    $80    ; 94  e
     9FA2   10             6783            .byte    $10    ; 95  e
     9FA3   C0             6784            .byte    $c0    ; 96  e
     9FA4   50             6785            .byte    $50    ; 97  e
     9FA5   30             6786            .byte    $30    ; 98  a
     9FA6   30             6787            .byte    $30    ; 99  e
     9FA7   F0             6788            .byte    $f0    ; 9a  e
     9FA8   70             6789            .byte    $70    ; 9b  e
     9FA9   90             6790            .byte    $90    ; 9c  e
     9FAA   B0             6791            .byte    $b0    ; 9d  e
     9FAB   D0             6792            .byte    $d0    ; 9e  e
     9FAC   FF             6793            .byte    $ff    ; 9f
     9FAD   FF             6794            .byte    $ff    ; a0
     9FAE   FF             6795            .byte    $ff    ; a1
     9FAF   00             6796            .byte    $00    ; a2  d
     9FB0   0A             6797            .byte    $0a    ; a3  d
     9FB1   FF             6798            .byte    $ff    ; a4
     9FB2   FF             6799            .byte    $ff    ; a5
     9FB3   FF             6800            .byte    $ff    ; a6
     9FB4   FF             6801            .byte    $ff    ; a7
     9FB5   F0             6802            .byte    $f0    ; a8  a
                           6803
     9FB6   00             6804   stbctl   brk              ; a9  *** start code stbctl
     9FB7   EA             6805            nop              ; aa
     9FB8   B5 00          6806   stbknt   lda  $00,x       ; ab,ac
     9FBA   30 FC          6807            bmi  stbknt      ; ad,ae
     9FBC   60             6808            rts              ; af  *** end code stbctl
                           6809
     9FBD   60             6810            .byte    $60    ; b0  a
     9FBE   FF             6811            .byte    $ff    ; b1
     9FBF   01             6812            .byte    $01    ; b2  d
     9FC0   0B             6813            .byte    $0b    ; b3  d
     9FC1   FF             6814            .byte    $ff    ; b4
     9FC2   FF             6815            .byte    $ff    ; b5
     9FC3   FF             6816            .byte    $ff    ; b6
```

```
Error Addr  Code           Seq    Source statement

     9FC4   FF             6817            .byte   $ff     ; b7
     9FC5   70             6818            .byte   $70     ; b8  a
     9FC6   FF             6819            .byte   $ff     ; b9
     9FC7   FF             6820            .byte   $ff     ; ba
     9FC8   FF             6821            .byte   $ff     ; bb
     9FC9   FF             6822            .byte   $ff     ; bc
     9FCA   FF             6823            .byte   $ff     ; bd
     9FCB   C0             6824            .byte   $c0     ; be  g
     9FCC   F0             6825            .byte   $f0     ; bf  g
     9FCD   D0             6826            .byte   $d0     ; c0  g
     9FCE   FF             6827            .byte   $ff     ; c1
     9FCF   01             6828            .byte   $01     ; c2  b
     9FD0   05             6829            .byte   $05     ; c3  b
     9FD1   03             6830            .byte   $03     ; c4  b
     9FD2   07             6831            .byte   $07     ; c5  b
     9FD3   0B             6832            .byte   $0b     ; c6  b
     9FD4   FF             6833            .byte   $ff     ; c7
     9FD5   90             6834            .byte   $90     ; c8  a
     9FD6   FF             6835            .byte   $ff     ; c9
     9FD7   FF             6836            .byte   $ff     ; ca
     9FD8   FF             6837            .byte   $ff     ; cb
     9FD9   FF             6838            .byte   $ff     ; cc
     9FDA   FF             6839            .byte   $ff     ; cd
     9FDB   FF             6840            .byte   $ff     ; ce
     9FDC   FF             6841            .byte   $ff     ; cf
     9FDD   A0             6842            .byte   $a0     ; d0  a
     9FDE   FF             6843            .byte   $ff     ; d1
     9FDF   0C             6844            .byte   $0c     ; d2  d
     9FE0   0D             6845            .byte   $0d     ; d3  d
     9FE1   FF             6846            .byte   $ff     ; d4
     9FE2   FF             6847            .byte   $ff     ; d5
     9FE3   FF             6848            .byte   $ff     ; d6
     9FE4   FF             6849            .byte   $ff     ; d7
     9FE5   B0             6850            .byte   $b0     ; d8  a
     9FE6   FF             6851            .byte   $ff     ; d9
     9FE7   FF             6852            .byte   $ff     ; da
     9FE8   FF             6853            .byte   $ff     ; db
     9FE9   FF             6854            .byte   $ff     ; dc
     9FEA   FF             6855            .byte   $ff     ; dd
     9FEB   40             6856            .byte   $40     ; de  g
     9FEC   60             6857            .byte   $60     ; df  g
     9FED   E0             6858            .byte   $e0     ; e0  g
     9FEE   FF             6859            .byte   $ff     ; e1
     9FEF   04             6860            .byte   $04     ; e2  d
     9FF0   0E             6861            .byte   $0e     ; e3  d
     9FF1   FF             6862            .byte   $ff     ; e4
     9FF2   FF             6863            .byte   $ff     ; e5
     9FF3   FF             6864            .byte   $ff     ; e6
     9FF4   FF             6865            .byte   $ff     ; e7
     9FF5   D0             6866            .byte   $d0     ; e8  a
```

```
Error Addr  Code           Seq    Source statement

     9FF6   FF             6867           .byte    $ff     ; e9
     9FF7   FF             6868           .byte    $ff     ; ea
     9FF8   FF             6869           .byte    $ff     ; eb
     9FF9   FF             6870           .byte    $ff     ; ec
     9FFA   FF             6871           .byte    $ff     ; ed
     9FFB   FF             6872           .byte    $ff     ; ee
     9FFC   FF             6873           .byte    $ff     ; ef
     9FFD   E0             6874           .byte    $e0     ; f0  a
     9FFE   FF             6875           .byte    $ff     ; f1
     9FFF   05             6876           .byte    $05     ; f2  d
     A000   FF             6877           .byte    $ff     ; f3
     A001   FF             6878           .byte    $ff     ; f4
     A002   FF             6879           .byte    $ff     ; f5
     A003   FF             6880           .byte    $ff     ; f6
     A004   FF             6881           .byte    $ff     ; f7
     A005   FF             6882           .byte    $ff     ; f8
     A006   FF             6883           .byte    $ff     ; f9
     A007   FF             6884           .byte    $ff     ; fa
     A008   FF             6885           .byte    $ff     ; fb
     A009   FF             6886           .byte    $ff     ; fc
     A00A   FF             6887           .byte    $ff     ; fd
     A00B   50             6888           .byte    $50     ; fe  g
     A00C   70             6889           .byte    $70     ; ff  g
                           6890           .include  gcrtable1
```

```
Error Addr  Code           Seq    Source statement

                           6892   ;       10/25/84 ds
                           6893
            A00D   0C      6894   gcrtb1  .byte   $0c     ; 0
            A00E   04      6895           .byte   $04     ; 1
            A00F   05      6896           .byte   $05     ; 2
            A010   FF      6897           .byte   $ff     ; 3
            A011   FF      6898           .byte   $ff     ; 4
            A012   02      6899           .byte   $02     ; 5
            A013   03      6900           .byte   $03     ; 6
            A014   FF      6901           .byte   $ff     ; 7
            A015   0F      6902           .byte   $0f     ; 8
            A016   06      6903           .byte   $06     ; 9
            A017   07      6904           .byte   $07     ; a
            A018   FF      6905           .byte   $ff     ; b
            A019   09      6906           .byte   $09     ; c
            A01A   0A      6907           .byte   $0a     ; d
            A01B   0B      6908           .byte   $0b     ; e
            A01C   FF      6909           .byte   $ff     ; f
            A01D   0D      6910           .byte   $0d     ; 10
            A01E   0E      6911           .byte   $0e     ; 11
            A01F   80      6912           .byte   $80     ; 12
            A020   FF      6913           .byte   $ff     ; 13
            A021   00      6914           .byte   $00     ; 14
            A022   00      6915           .byte   $00     ; 15
            A023   10      6916           .byte   $10     ; 16
            A024   40      6917           .byte   $40     ; 17
            A025   FF      6918           .byte   $ff     ; 18
            A026   20      6919           .byte   $20     ; 19
            A027   C0      6920           .byte   $c0     ; 1a
            A028   60      6921           .byte   $60     ; 1b
            A029   40      6922           .byte   $40     ; 1c
            A02A   A0      6923           .byte   $a0     ; 1d
            A02B   50      6924           .byte   $50     ; 1e
            A02C   E0      6925           .byte   $e0     ; 1f
            A02D   FF      6926           .byte   $ff     ; 20
            A02E   FF      6927           .byte   $ff     ; 21
            A02F   FF      6928           .byte   $ff     ; 22
            A030   02      6929           .byte   $02     ; 23
            A031   20      6930           .byte   $20     ; 24
            A032   08      6931           .byte   $08     ; 25
            A033   30      6932           .byte   $30     ; 26
            A034   30      6933           .byte   $30     ; 27
            A035   30      6934           .byte   $30     ; 28
            A036   00      6935           .byte   $00     ; 29
            A037   F0      6936           .byte   $f0     ; 2a
            A038   FF      6937           .byte   $ff     ; 2b
            A039   60      6938           .byte   $60     ; 2c
            A03A   01      6939           .byte   $01     ; 2d
            A03B   70      6940           .byte   $70     ; 2e
            A03C   FF      6941           .byte   $ff     ; 2f
```

```
Error Addr  Code           Seq    Source statement

     A03D   FF            6942            .byte   $ff     ; 30
     A03E   FF            6943            .byte   $ff     ; 31
     A03F   90            6944            .byte   $90     ; 32
     A040   03            6945            .byte   $03     ; 33
     A041   A0            6946            .byte   $a0     ; 34
     A042   0C            6947            .byte   $0c     ; 35
     A043   B0            6948            .byte   $b0     ; 36
     A044   FF            6949            .byte   $ff     ; 37
     A045   FF            6950            .byte   $ff     ; 38
     A046   04            6951            .byte   $04     ; 39
     A047   D0            6952            .byte   $d0     ; 3a
     A048   FF            6953            .byte   $ff     ; 3b
     A049   E0            6954            .byte   $e0     ; 3c
     A04A   05            6955            .byte   $05     ; 3d
     A04B   80            6956            .byte   $80     ; 3e
     A04C   FF            6957            .byte   $ff     ; 3f
     A04D   90            6958            .byte   $90     ; 40
     A04E   FF            6959            .byte   $ff     ; 41
     A04F   08            6960            .byte   $08     ; 42
     A050   0C            6961            .byte   $0c     ; 43
     A051   FF            6962            .byte   $ff     ; 44
     A052   0F            6963            .byte   $0f     ; 45
     A053   09            6964            .byte   $09     ; 46
     A054   0D            6965            .byte   $0d     ; 47
                          6966
     A055   80            6967            .byte   $80     ; 48
     A056   80            6968            .byte   $80     ; 49
     A057   80            6969            .byte   $80     ; 4a
     A058   80            6970            .byte   $80     ; 4b
     A059   80            6971            .byte   $80     ; 4c
     A05A   80            6972            .byte   $80     ; 4d
     A05B   80            6973            .byte   $80     ; 4e
     A05C   80            6974            .byte   $80     ; 4f
                          6975
     A05D   00            6976            .byte   $00     ; 50
     A05E   00            6977            .byte   $00     ; 51
     A05F   00            6978            .byte   $00     ; 52
     A060   00            6979            .byte   $00     ; 53
     A061   00            6980            .byte   $00     ; 54
     A062   00            6981            .byte   $00     ; 55
     A063   00            6982            .byte   $00     ; 56
     A064   00            6983            .byte   $00     ; 57
                          6984
     A065   10            6985            .byte   $10     ; 58
     A066   10            6986            .byte   $10     ; 59
     A067   10            6987            .byte   $10     ; 5a
     A068   10            6988            .byte   $10     ; 5b
     A069   10            6989            .byte   $10     ; 5c
     A06A   10            6990            .byte   $10     ; 5d
     A06B   10            6991            .byte   $10     ; 5e
```

```
Error Addr  Code          Seq    Source statement

     A06C   10          6992            .byte   $10     ; 5f
                        6993
     A06D   A0          6994            .byte   $a0     ; 60
     A06E   FF          6995            .byte   $ff     ; 61
     A06F   FF          6996            .byte   $ff     ; 62
     A070   06          6997            .byte   $06     ; 63
     A071   FF          6998            .byte   $ff     ; 64
     A072   09          6999            .byte   $09     ; 65
     A073   FF          7000            .byte   $ff     ; 66
     A074   FF          7001            .byte   $ff     ; 67
                        7002
     A075   C0          7003            .byte   $c0     ; 68
     A076   C0          7004            .byte   $c0     ; 69
     A077   C0          7005            .byte   $c0     ; 6a
     A078   C0          7006            .byte   $c0     ; 6b
     A079   C0          7007            .byte   $c0     ; 6c
     A07A   C0          7008            .byte   $c0     ; 6d
     A07B   C0          7009            .byte   $c0     ; 6e
     A07C   C0          7010            .byte   $c0     ; 6f
                        7011
     A07D   40          7012            .byte   $40     ; 70
     A07E   40          7013            .byte   $40     ; 71
     A07F   40          7014            .byte   $40     ; 72
     A080   40          7015            .byte   $40     ; 73
     A081   40          7016            .byte   $40     ; 74
     A082   40          7017            .byte   $40     ; 75
     A083   40          7018            .byte   $40     ; 76
     A084   40          7019            .byte   $40     ; 77
                        7020
     A085   50          7021            .byte   $50     ; 78
     A086   50          7022            .byte   $50     ; 79
     A087   50          7023            .byte   $50     ; 7a
     A088   50          7024            .byte   $50     ; 7b
     A089   50          7025            .byte   $50     ; 7c
     A08A   50          7026            .byte   $50     ; 7d
     A08B   50          7027            .byte   $50     ; 7e
     A08C   50          7028            .byte   $50     ; 7f
                        7029
     A08D   B0          7030            .byte   $b0     ; 80
     A08E   FF          7031            .byte   $ff     ; 81
     A08F   00          7032            .byte   $00     ; 82
     A090   04          7033            .byte   $04     ; 83
     A091   02          7034            .byte   $02     ; 84
     A092   06          7035            .byte   $06     ; 85
     A093   0A          7036            .byte   $0a     ; 86
     A094   0E          7037            .byte   $0e     ; 87
                        7038
     A095   80          7039            .byte   $80     ; 88
     A096   80          7040            .byte   $80     ; 89
     A097   80          7041            .byte   $80     ; 8a
```

```
Error Addr  Code           Seq    Source statement

     A098   80            7042            .byte   $80     ; 8b
     A099   80            7043            .byte   $80     ; 8c
     A09A   80            7044            .byte   $80     ; 8d
     A09B   80            7045            .byte   $80     ; 8e
     A09C   80            7046            .byte   $80     ; 8f
                          7047
     A09D   20            7048            .byte   $20     ; 90
     A09E   20            7049            .byte   $20     ; 91
     A09F   20            7050            .byte   $20     ; 92
     A0A0   20            7051            .byte   $20     ; 93
     A0A1   20            7052            .byte   $20     ; 94
     A0A2   20            7053            .byte   $20     ; 95
     A0A3   20            7054            .byte   $20     ; 96
     A0A4   20            7055            .byte   $20     ; 97
                          7056
     A0A5   30            7057            .byte   $30     ; 98
     A0A6   30            7058            .byte   $30     ; 99
     A0A7   30            7059            .byte   $30     ; 9a
     A0A8   30            7060            .byte   $30     ; 9b
     A0A9   30            7061            .byte   $30     ; 9c
     A0AA   30            7062            .byte   $30     ; 9d
     A0AB   30            7063            .byte   $30     ; 9e
     A0AC   30            7064            .byte   $30     ; 9f
                          7065
     A0AD   FF            7066            .byte   $ff     ; a0
     A0AE   FF            7067            .byte   $ff     ; a1
     A0AF   00            7068            .byte   $00     ; a2
     A0B0   0A            7069            .byte   $0a     ; a3
     A0B1   0A            7070            .byte   $0a     ; a4
     A0B2   0A            7071            .byte   $0a     ; a5
     A0B3   0A            7072            .byte   $0a     ; a6
     A0B4   0A            7073            .byte   $0a     ; a7
                          7074
     A0B5   F0            7075            .byte   $f0     ; a8
     A0B6   F0            7076            .byte   $f0     ; a9
     A0B7   F0            7077            .byte   $f0     ; aa
     A0B8   F0            7078            .byte   $f0     ; ab
     A0B9   F0            7079            .byte   $f0     ; ac
     A0BA   F0            7080            .byte   $f0     ; ad
     A0BB   F0            7081            .byte   $f0     ; ae
     A0BC   F0            7082            .byte   $f0     ; af
                          7083
     A0BD   60            7084            .byte   $60     ; b0
     A0BE   60            7085            .byte   $60     ; b1
     A0BF   60            7086            .byte   $60     ; b2
     A0C0   60            7087            .byte   $60     ; b3
     A0C1   60            7088            .byte   $60     ; b4
     A0C2   60            7089            .byte   $60     ; b5
     A0C3   60            7090            .byte   $60     ; b6
     A0C4   60            7091            .byte   $60     ; b7
```

```
Error Addr  Code           Seq    Source statement

                           7092
      AOC5  70             7093            .byte   $70     ; b8
      AOC6  70             7094            .byte   $70     ; b9
      AOC7  70             7095            .byte   $70     ; ba
      AOC8  70             7096            .byte   $70     ; bb
      AOC9  70             7097            .byte   $70     ; bc
      AOCA  70             7098            .byte   $70     ; bd
      AOCB  70             7099            .byte   $70     ; be
      AOCC  70             7100            .byte   $70     ; bf
                           7101
      AOCD  D0             7102            .byte   $d0     ; c0
      AOCE  FF             7103            .byte   $ff     ; c1
      AOCF  01             7104            .byte   $01     ; c2
      AOD0  05             7105            .byte   $05     ; c3
      AOD1  03             7106            .byte   $03     ; c4
      AOD2  07             7107            .byte   $07     ; c5
      AOD3  0B             7108            .byte   $0b     ; c6
      AOD4  FF             7109            .byte   $ff     ; c7
                           7110
      AOD5  90             7111            .byte   $90     ; c8
      AOD6  90             7112            .byte   $90     ; c9
      AOD7  90             7113            .byte   $90     ; ca
      AOD8  90             7114            .byte   $90     ; cb
      AOD9  90             7115            .byte   $90     ; cc
      AODA  90             7116            .byte   $90     ; cd
      AODB  90             7117            .byte   $90     ; ce
      AODC  90             7118            .byte   $90     ; cf
                           7119
      AODD  A0             7120            .byte   $a0     ; d0
      AODE  A0             7121            .byte   $a0     ; d1
      AODF  A0             7122            .byte   $a0     ; d2
      AOE0  A0             7123            .byte   $a0     ; d3
      AOE1  A0             7124            .byte   $a0     ; d4
      AOE2  A0             7125            .byte   $a0     ; d5
      AOE3  A0             7126            .byte   $a0     ; d6
      AOE4  A0             7127            .byte   $a0     ; d7
                           7128
      AOE5  B0             7129            .byte   $b0     ; d8
      AOE6  B0             7130            .byte   $b0     ; d9
      AOE7  B0             7131            .byte   $b0     ; da
      AOE8  B0             7132            .byte   $b0     ; db
      AOE9  B0             7133            .byte   $b0     ; dc
      AOEA  B0             7134            .byte   $b0     ; dd
      AOEB  B0             7135            .byte   $b0     ; de
      AOEC  B0             7136            .byte   $b0     ; df
                           7137
      AOED  E0             7138            .byte   $e0     ; e0
      AOEE  FF             7139            .byte   $ff     ; e1
      AOEF  04             7140            .byte   $04     ; e2
      AOF0  0E             7141            .byte   $0e     ; e3
```

```
Error Addr  Code           Seq    Source statement

      AOF1  FF            7142            .byte   $ff     ; e4
      AOF2  FF            7143            .byte   $ff     ; e5
      AOF3  FF            7144            .byte   $ff     ; e6
      AOF4  FF            7145            .byte   $ff     ; e7
                          7146
      AOF5  D0            7147            .byte   $d0     ; e8
      AOF6  D0            7148            .byte   $d0     ; e9
      AOF7  D0            7149            .byte   $d0     ; ea
      AOF8  D0            7150            .byte   $d0     ; eb
      AOF9  D0            7151            .byte   $d0     ; ec
      AOFA  D0            7152            .byte   $d0     ; ed
      AOFB  D0            7153            .byte   $d0     ; ee
      AOFC  D0            7154            .byte   $d0     ; ef
                          7155
      AOFD  E0            7156            .byte   $e0     ; f0
      AOFE  E0            7157            .byte   $e0     ; f1
      AOFF  E0            7158            .byte   $e0     ; f2
      A100  E0            7159            .byte   $e0     ; f3
      A101  E0            7160            .byte   $e0     ; f4
      A102  E0            7161            .byte   $e0     ; f5
      A103  E0            7162            .byte   $e0     ; f6
      A104  E0            7163            .byte   $e0     ; f7
                          7164
      A105  05            7165            .byte   $05     ; f8
      A106  05            7166            .byte   $05     ; f9
      A107  05            7167            .byte   $05     ; fa
      A108  05            7168            .byte   $05     ; fb
      A109  05            7169            .byte   $05     ; fc
      A10A  05            7170            .byte   $05     ; fd
      A10B  50            7171            .byte   $50     ; fe
      A10C  70            7172            .byte   $70     ; ff
                          7173            .include  gcrtable3
```

```
Error Addr  Code           Seq    Source statement

                           7175   ;       11/14/84 ds
                           7176
            A10D   FF      7177   gcrtb2  .byte   $ff     ; 0
            A10E   FF      7178           .byte   $ff     ; 1
            A10F   FF      7179           .byte   $ff     ; 2
            A110   FF      7180           .byte   $ff     ; 3
            A111   FF      7181           .byte   $ff     ; 4
            A112   FF      7182           .byte   $ff     ; 5
            A113   FF      7183           .byte   $ff     ; 6
            A114   FF      7184           .byte   $ff     ; 7
            A115   FF      7185           .byte   $ff     ; 8
            A116   FF      7186           .byte   $ff     ; 9
            A117   FF      7187           .byte   $ff     ; a
            A118   FF      7188           .byte   $ff     ; b
            A119   FF      7189           .byte   $ff     ; c
            A11A   FF      7190           .byte   $ff     ; d
            A11B   FF      7191           .byte   $ff     ; e
            A11C   FF      7192           .byte   $ff     ; f
            A11D   FF      7193           .byte   $ff     ; 10
            A11E   FF      7194           .byte   $ff     ; 11
            A11F   80      7195           .byte   $80     ; 12
            A120   80      7196           .byte   $80     ; 13
            A121   00      7197           .byte   $00     ; 14
            A122   00      7198           .byte   $00     ; 15
            A123   10      7199           .byte   $10     ; 16
            A124   10      7200           .byte   $10     ; 17
            A125   FF      7201           .byte   $ff     ; 18
            A126   FF      7202           .byte   $ff     ; 19
            A127   C0      7203           .byte   $c0     ; 1a
            A128   C0      7204           .byte   $c0     ; 1b
            A129   40      7205           .byte   $40     ; 1c
            A12A   40      7206           .byte   $40     ; 1d
            A12B   50      7207           .byte   $50     ; 1e
            A12C   50      7208           .byte   $50     ; 1f
            A12D   FF      7209           .byte   $ff     ; 20
            A12E   FF      7210           .byte   $ff     ; 21
            A12F   FF      7211           .byte   $ff     ; 22
            A130   FF      7212           .byte   $ff     ; 23
            A131   20      7213           .byte   $20     ; 24
            A132   20      7214           .byte   $20     ; 25
            A133   30      7215           .byte   $30     ; 26
            A134   30      7216           .byte   $30     ; 27
            A135   FF      7217           .byte   $ff     ; 28
            A136   FF      7218           .byte   $ff     ; 29
            A137   F0      7219           .byte   $f0     ; 2a
            A138   F0      7220           .byte   $f0     ; 2b
            A139   60      7221           .byte   $60     ; 2c
            A13A   60      7222           .byte   $60     ; 2d
            A13B   70      7223           .byte   $70     ; 2e
            A13C   70      7224           .byte   $70     ; 2f
```

```
Error Addr  Code           Seq    Source statement

     A13D   FF             7225            .byte   $ff     ; 30
     A13E   FF             7226            .byte   $ff     ; 31
     A13F   90             7227            .byte   $90     ; 32
     A140   90             7228            .byte   $90     ; 33
     A141   A0             7229            .byte   $a0     ; 34
     A142   A0             7230            .byte   $a0     ; 35
     A143   B0             7231            .byte   $b0     ; 36
     A144   B0             7232            .byte   $b0     ; 37
     A145   FF             7233            .byte   $ff     ; 38
     A146   FF             7234            .byte   $ff     ; 39
     A147   D0             7235            .byte   $d0     ; 3a
     A148   D0             7236            .byte   $d0     ; 3b
     A149   E0             7237            .byte   $e0     ; 3c
     A14A   E0             7238            .byte   $e0     ; 3d
     A14B   FF             7239            .byte   $ff     ; 3e
     A14C   FF             7240            .byte   $ff     ; 3f
     A14D   FF             7241            .byte   $ff     ; 40
     A14E   FF             7242            .byte   $ff     ; 41
     A14F   FF             7243            .byte   $ff     ; 42
     A150   FF             7244            .byte   $ff     ; 43
     A151   FF             7245            .byte   $ff     ; 44
     A152   FF             7246            .byte   $ff     ; 45
     A153   FF             7247            .byte   $ff     ; 46
     A154   FF             7248            .byte   $ff     ; 47
     A155   FF             7249            .byte   $ff     ; 48
     A156   FF             7250            .byte   $ff     ; 49
     A157   FF             7251            .byte   $ff     ; 4a
     A158   FF             7252            .byte   $ff     ; 4b
     A159   FF             7253            .byte   $ff     ; 4c
     A15A   FF             7254            .byte   $ff     ; 4d
     A15B   FF             7255            .byte   $ff     ; 4e
     A15C   FF             7256            .byte   $ff     ; 4f
                           7257
     A15D   FF             7258            .byte   $ff     ; 50
     A15E   FF             7259            .byte   $ff     ; 51
     A15F   80             7260            .byte   $80     ; 52
     A160   80             7261            .byte   $80     ; 53
     A161   00             7262            .byte   $00     ; 54
     A162   00             7263            .byte   $00     ; 55
     A163   10             7264            .byte   $10     ; 56
     A164   10             7265            .byte   $10     ; 57
                           7266
     A165   FF             7267            .byte   $ff     ; 58
     A166   FF             7268            .byte   $ff     ; 59
     A167   C0             7269            .byte   $c0     ; 5a
     A168   C0             7270            .byte   $c0     ; 5b
     A169   40             7271            .byte   $40     ; 5c
     A16A   40             7272            .byte   $40     ; 5d
     A16B   50             7273            .byte   $50     ; 5e
     A16C   50             7274            .byte   $50     ; 5f
```

```
Error Addr  Code           Seq    Source statement

                          7275
      A16D  FF            7276            .byte   $ff     ; 60
      A16E  FF            7277            .byte   $ff     ; 61
      A16F  FF            7278            .byte   $ff     ; 62
      A170  FF            7279            .byte   $ff     ; 63
      A171  20            7280            .byte   $20     ; 64
      A172  20            7281            .byte   $20     ; 65
      A173  30            7282            .byte   $30     ; 66
      A174  30            7283            .byte   $30     ; 67
                          7284
      A175  FF            7285            .byte   $ff     ; 68
      A176  FF            7286            .byte   $ff     ; 69
      A177  F0            7287            .byte   $f0     ; 6a
      A178  F0            7288            .byte   $f0     ; 6b
      A179  60            7289            .byte   $60     ; 6c
      A17A  60            7290            .byte   $60     ; 6d
      A17B  70            7291            .byte   $70     ; 6e
      A17C  70            7292            .byte   $70     ; 6f
                          7293
      A17D  FF            7294            .byte   $ff     ; 70
      A17E  FF            7295            .byte   $ff     ; 71
      A17F  90            7296            .byte   $90     ; 72
      A180  90            7297            .byte   $90     ; 73
      A181  A0            7298            .byte   $a0     ; 74
      A182  A0            7299            .byte   $a0     ; 75
      A183  B0            7300            .byte   $b0     ; 76
      A184  B0            7301            .byte   $b0     ; 77
                          7302
      A185  FF            7303            .byte   $ff     ; 78
      A186  FF            7304            .byte   $ff     ; 79
      A187  D0            7305            .byte   $d0     ; 7a
      A188  D0            7306            .byte   $d0     ; 7b
      A189  E0            7307            .byte   $e0     ; 7c
      A18A  E0            7308            .byte   $e0     ; 7d
      A18B  FF            7309            .byte   $ff     ; 7e
      A18C  FF            7310            .byte   $ff     ; 7f
                          7311
      A18D  FF            7312            .byte   $ff     ; 80
      A18E  FF            7313            .byte   $ff     ; 81
      A18F  FF            7314            .byte   $ff     ; 82
      A190  FF            7315            .byte   $ff     ; 83
      A191  FF            7316            .byte   $ff     ; 84
      A192  FF            7317            .byte   $ff     ; 85
      A193  FF            7318            .byte   $ff     ; 86
      A194  FF            7319            .byte   $ff     ; 87
                          7320
      A195  FF            7321            .byte   $ff     ; 88
      A196  FF            7322            .byte   $ff     ; 89
      A197  FF            7323            .byte   $ff     ; 8a
      A198  FF            7324            .byte   $ff     ; 8b
```

```
Error Addr  Code          Seq    Source statement

     A199   FF           7325           .byte   $ff     ; 8c
     A19A   FF           7326           .byte   $ff     ; 8d
     A19B   FF           7327           .byte   $ff     ; 8e
     A19C   FF           7328           .byte   $ff     ; 8f
                         7329
     A19D   FF           7330           .byte   $ff     ; 90
     A19E   FF           7331           .byte   $ff     ; 91
     A19F   80           7332           .byte   $80     ; 92
     A1A0   80           7333           .byte   $80     ; 93
     A1A1   00           7334           .byte   $00     ; 94
     A1A2   00           7335           .byte   $00     ; 95
     A1A3   10           7336           .byte   $10     ; 96
     A1A4   10           7337           .byte   $10     ; 97
                         7338
     A1A5   FF           7339           .byte   $ff     ; 98
     A1A6   FF           7340           .byte   $ff     ; 99
     A1A7   C0           7341           .byte   $c0     ; 9a
     A1A8   C0           7342           .byte   $c0     ; 9b
     A1A9   40           7343           .byte   $40     ; 9c
     A1AA   40           7344           .byte   $40     ; 9d
     A1AB   50           7345           .byte   $50     ; 9e
     A1AC   50           7346           .byte   $50     ; 9f
                         7347
     A1AD   FF           7348           .byte   $ff     ; a0
     A1AE   FF           7349           .byte   $ff     ; a1
     A1AF   FF           7350           .byte   $ff     ; a2
     A1B0   FF           7351           .byte   $ff     ; a3
     A1B1   20           7352           .byte   $20     ; a4
     A1B2   20           7353           .byte   $20     ; a5
     A1B3   30           7354           .byte   $30     ; a6
     A1B4   30           7355           .byte   $30     ; a7
                         7356
     A1B5   FF           7357           .byte   $ff     ; a8
     A1B6   FF           7358           .byte   $ff     ; a9
     A1B7   F0           7359           .byte   $f0     ; aa
     A1B8   F0           7360           .byte   $f0     ; ab
     A1B9   60           7361           .byte   $60     ; ac
     A1BA   60           7362           .byte   $60     ; ad
     A1BB   70           7363           .byte   $70     ; ae
     A1BC   70           7364           .byte   $70     ; af
                         7365
     A1BD   FF           7366           .byte   $ff     ; b0
     A1BE   FF           7367           .byte   $ff     ; b1
     A1BF   90           7368           .byte   $90     ; b2
     A1C0   90           7369           .byte   $90     ; b3
     A1C1   A0           7370           .byte   $a0     ; b4
     A1C2   A0           7371           .byte   $a0     ; b5
     A1C3   B0           7372           .byte   $b0     ; b6
     A1C4   B0           7373           .byte   $b0     ; b7
                         7374
```

```
Error Addr  Code          Seq    Source statement

      A1C5  FF            7375           .byte   $ff     ; b8
      A1C6  FF            7376           .byte   $ff     ; b9
      A1C7  D0            7377           .byte   $d0     ; ba
      A1C8  D0            7378           .byte   $d0     ; bb
      A1C9  E0            7379           .byte   $e0     ; bc
      A1CA  E0            7380           .byte   $e0     ; bd
      A1CB  FF            7381           .byte   $ff     ; be
      A1CC  FF            7382           .byte   $ff     ; bf
                          7383
      A1CD  FF            7384           .byte   $ff     ; c0
      A1CE  FF            7385           .byte   $ff     ; c1
      A1CF  FF            7386           .byte   $ff     ; c2
      A1D0  FF            7387           .byte   $ff     ; c3
      A1D1  FF            7388           .byte   $ff     ; c4
      A1D2  FF            7389           .byte   $ff     ; c5
      A1D3  FF            7390           .byte   $ff     ; c6
      A1D4  FF            7391           .byte   $ff     ; c7
                          7392
      A1D5  FF            7393           .byte   $ff     ; c8
      A1D6  FF            7394           .byte   $ff     ; c9
      A1D7  FF            7395           .byte   $ff     ; ca
      A1D8  FF            7396           .byte   $ff     ; cb
      A1D9  FF            7397           .byte   $ff     ; cc
      A1DA  FF            7398           .byte   $ff     ; cd
      A1DB  FF            7399           .byte   $ff     ; ce
      A1DC  FF            7400           .byte   $ff     ; cf
                          7401
      A1DD  FF            7402           .byte   $ff     ; d0
      A1DE  FF            7403           .byte   $ff     ; d1
      A1DF  80            7404           .byte   $80     ; d2
      A1E0  80            7405           .byte   $80     ; d3
      A1E1  00            7406           .byte   $00     ; d4
      A1E2  00            7407           .byte   $00     ; d5
      A1E3  10            7408           .byte   $10     ; d6
      A1E4  10            7409           .byte   $10     ; d7
                          7410
      A1E5  FF            7411           .byte   $ff     ; d8
      A1E6  FF            7412           .byte   $ff     ; d9
      A1E7  C0            7413           .byte   $c0     ; da
      A1E8  C0            7414           .byte   $c0     ; db
      A1E9  40            7415           .byte   $40     ; dc
      A1EA  40            7416           .byte   $40     ; dd
      A1EB  50            7417           .byte   $50     ; de
      A1EC  50            7418           .byte   $50     ; df
                          7419
      A1ED  FF            7420           .byte   $ff     ; e0
      A1EE  FF            7421           .byte   $ff     ; e1
      A1EF  FF            7422           .byte   $ff     ; e2
      A1F0  FF            7423           .byte   $ff     ; e3
      A1F1  20            7424           .byte   $20     ; e4
```

```
Error Addr  Code           Seq    Source statement

     A1F2   20             7425           .byte    $20    ; e5
     A1F3   30             7426           .byte    $30    ; e6
     A1F4   30             7427           .byte    $30    ; e7
                           7428
     A1F5   FF             7429           .byte    $ff    ; e8
     A1F6   FF             7430           .byte    $ff    ; e9
     A1F7   F0             7431           .byte    $f0    ; ea
     A1F8   F0             7432           .byte    $f0    ; eb
     A1F9   60             7433           .byte    $60    ; ec
     A1FA   60             7434           .byte    $60    ; ed
     A1FB   70             7435           .byte    $70    ; ee
     A1FC   70             7436           .byte    $70    ; ef
                           7437
     A1FD   FF             7438           .byte    $ff    ; f0
     A1FE   FF             7439           .byte    $ff    ; f1
     A1FF   90             7440           .byte    $90    ; f2
     A200   90             7441           .byte    $90    ; f3
     A201   A0             7442           .byte    $a0    ; f4
     A202   A0             7443           .byte    $a0    ; f5
     A203   B0             7444           .byte    $b0    ; f6
     A204   B0             7445           .byte    $b0    ; f7
                           7446
     A205   FF             7447           .byte    $ff    ; f8
     A206   FF             7448           .byte    $ff    ; f9
     A207   D0             7449           .byte    $d0    ; fa
     A208   D0             7450           .byte    $d0    ; fb
     A209   E0             7451           .byte    $e0    ; fc
     A20A   E0             7452           .byte    $e0    ; fd
     A20B   FF             7453           .byte    $ff    ; fe
     A20C   FF             7454           .byte    $ff    ; ff
                           7455           .include  gcrtable6
```

```
Error Addr  Code           Seq    Source statement

                           7457   ;       10/25/84 ds
                           7458
     A20D   FF             7459   gcrtb3  .byte    $ff     ; 0
     A20E   FF             7460           .byte    $ff     ; 1
     A20F   FF             7461           .byte    $ff     ; 2
     A210   FF             7462           .byte    $ff     ; 3
     A211   FF             7463           .byte    $ff     ; 4
     A212   FF             7464           .byte    $ff     ; 5
     A213   FF             7465           .byte    $ff     ; 6
     A214   FF             7466           .byte    $ff     ; 7
     A215   FF             7467           .byte    $ff     ; 8
     A216   FF             7468           .byte    $ff     ; 9
     A217   FF             7469           .byte    $ff     ; a
     A218   FF             7470           .byte    $ff     ; b
     A219   FF             7471           .byte    $ff     ; c
     A21A   FF             7472           .byte    $ff     ; d
     A21B   FF             7473           .byte    $ff     ; e
     A21C   FF             7474           .byte    $ff     ; f
     A21D   FF             7475           .byte    $ff     ; 10
     A21E   FF             7476           .byte    $ff     ; 11
     A21F   FF             7477           .byte    $ff     ; 12
     A220   FF             7478           .byte    $ff     ; 13
     A221   FF             7479           .byte    $ff     ; 14
     A222   FF             7480           .byte    $ff     ; 15
     A223   FF             7481           .byte    $ff     ; 16
     A224   FF             7482           .byte    $ff     ; 17
     A225   FF             7483           .byte    $ff     ; 18
     A226   FF             7484           .byte    $ff     ; 19
     A227   FF             7485           .byte    $ff     ; 1a
     A228   FF             7486           .byte    $ff     ; 1b
     A229   FF             7487           .byte    $ff     ; 1c
     A22A   FF             7488           .byte    $ff     ; 1d
     A22B   FF             7489           .byte    $ff     ; 1e
     A22C   FF             7490           .byte    $ff     ; 1f
     A22D   FF             7491           .byte    $ff     ; 20
     A22E   FF             7492           .byte    $ff     ; 21
     A22F   FF             7493           .byte    $ff     ; 22
     A230   FF             7494           .byte    $ff     ; 23
     A231   08             7495           .byte    $08     ; 24
     A232   08             7496           .byte    $08     ; 25
     A233   08             7497           .byte    $08     ; 26
     A234   08             7498           .byte    $08     ; 27
                           7499
     A235   00             7500           .byte    $00     ; 28
     A236   00             7501           .byte    $00     ; 29
     A237   00             7502           .byte    $00     ; 2a
     A238   00             7503           .byte    $00     ; 2b
     A239   01             7504           .byte    $01     ; 2c
     A23A   01             7505           .byte    $01     ; 2d
     A23B   01             7506           .byte    $01     ; 2e
```

```
Error Addr  Code          Seq    Source statement

     A23C   01           7507           .byte    $01    ; 2f
     A23D   FF           7508           .byte    $ff    ; 30
     A23E   FF           7509           .byte    $ff    ; 31
     A23F   FF           7510           .byte    $ff    ; 32
     A240   FF           7511           .byte    $ff    ; 33
     A241   0C           7512           .byte    $0c    ; 34
     A242   0C           7513           .byte    $0c    ; 35
     A243   0C           7514           .byte    $0c    ; 36
     A244   0C           7515           .byte    $0c    ; 37
     A245   04           7516           .byte    $04    ; 38
     A246   04           7517           .byte    $04    ; 39
     A247   04           7518           .byte    $04    ; 3a
     A248   04           7519           .byte    $04    ; 3b
     A249   05           7520           .byte    $05    ; 3c
     A24A   05           7521           .byte    $05    ; 3d
     A24B   05           7522           .byte    $05    ; 3e
     A24C   05           7523           .byte    $05    ; 3f
     A24D   FF           7524           .byte    $ff    ; 40
     A24E   FF           7525           .byte    $ff    ; 41
     A24F   FF           7526           .byte    $ff    ; 42
     A250   FF           7527           .byte    $ff    ; 43
     A251   FF           7528           .byte    $ff    ; 44
     A252   FF           7529           .byte    $ff    ; 45
     A253   FF           7530           .byte    $ff    ; 46
     A254   FF           7531           .byte    $ff    ; 47
     A255   02           7532           .byte    $02    ; 48
     A256   02           7533           .byte    $02    ; 49
     A257   02           7534           .byte    $02    ; 4a
     A258   02           7535           .byte    $02    ; 4b
     A259   03           7536           .byte    $03    ; 4c
     A25A   03           7537           .byte    $03    ; 4d
     A25B   03           7538           .byte    $03    ; 4e
     A25C   03           7539           .byte    $03    ; 4f
                         7540
     A25D   FF           7541           .byte    $ff    ; 50
     A25E   FF           7542           .byte    $ff    ; 51
     A25F   FF           7543           .byte    $ff    ; 52
     A260   FF           7544           .byte    $ff    ; 53
     A261   0F           7545           .byte    $0f    ; 54
     A262   0F           7546           .byte    $0f    ; 55
     A263   0F           7547           .byte    $0f    ; 56
     A264   0F           7548           .byte    $0f    ; 57
                         7549
     A265   06           7550           .byte    $06    ; 58
     A266   06           7551           .byte    $06    ; 59
     A267   06           7552           .byte    $06    ; 5a
     A268   06           7553           .byte    $06    ; 5b
     A269   07           7554           .byte    $07    ; 5c
     A26A   07           7555           .byte    $07    ; 5d
     A26B   07           7556           .byte    $07    ; 5e
```

```
Error Addr  Code          Seq    Source statement

     A26C   07           7557           .byte    $07     ; 5f
                         7558
     A26D   FF           7559           .byte    $ff     ; 60
     A26E   FF           7560           .byte    $ff     ; 61
     A26F   FF           7561           .byte    $ff     ; 62
     A270   FF           7562           .byte    $ff     ; 63
     A271   09           7563           .byte    $09     ; 64
     A272   09           7564           .byte    $09     ; 65
     A273   09           7565           .byte    $09     ; 66
     A274   09           7566           .byte    $09     ; 67
                         7567
     A275   0A           7568           .byte    $0a     ; 68
     A276   0A           7569           .byte    $0a     ; 69
     A277   0A           7570           .byte    $0a     ; 6a
     A278   0A           7571           .byte    $0a     ; 6b
     A279   0B           7572           .byte    $0b     ; 6c
     A27A   0B           7573           .byte    $0b     ; 6d
     A27B   0B           7574           .byte    $0b     ; 6e
     A27C   0B           7575           .byte    $0b     ; 6f
                         7576
     A27D   FF           7577           .byte    $ff     ; 70
     A27E   FF           7578           .byte    $ff     ; 71
     A27F   FF           7579           .byte    $ff     ; 72
     A280   FF           7580           .byte    $ff     ; 73
     A281   0D           7581           .byte    $0d     ; 74
     A282   0D           7582           .byte    $0d     ; 75
     A283   0D           7583           .byte    $0d     ; 76
     A284   0D           7584           .byte    $0d     ; 77
                         7585
     A285   0E           7586           .byte    $0e     ; 78
     A286   0E           7587           .byte    $0e     ; 79
     A287   0E           7588           .byte    $0e     ; 7a
     A288   0E           7589           .byte    $0e     ; 7b
     A289   FF           7590           .byte    $ff     ; 7c
     A28A   FF           7591           .byte    $ff     ; 7d
     A28B   FF           7592           .byte    $ff     ; 7e
     A28C   FF           7593           .byte    $ff     ; 7f
                         7594
     A28D   FF           7595           .byte    $ff     ; 80
     A28E   FF           7596           .byte    $ff     ; 81
     A28F   FF           7597           .byte    $ff     ; 82
     A290   FF           7598           .byte    $ff     ; 83
     A291   FF           7599           .byte    $ff     ; 84
     A292   FF           7600           .byte    $ff     ; 85
     A293   FF           7601           .byte    $ff     ; 86
     A294   FF           7602           .byte    $ff     ; 87
                         7603
     A295   FF           7604           .byte    $ff     ; 88
     A296   FF           7605           .byte    $ff     ; 89
     A297   FF           7606           .byte    $ff     ; 8a
```

```
Error Addr  Code          Seq .  Source statement

     A298   FF            7607           .byte    $ff     ; 8b
     A299   FF            7608           .byte    $ff     ; 8c
     A29A   FF            7609           .byte    $ff     ; 8d
     A29B   FF            7610           .byte    $ff     ; 8e
     A29C   FF            7611           .byte    $ff     ; 8f
                          7612
     A29D   FF            7613           .byte    $ff     ; 90
     A29E   FF            7614           .byte    $ff     ; 91
     A29F   FF            7615           .byte    $ff     ; 92
     A2A0   FF            7616           .byte    $ff     ; 93
     A2A1   FF            7617           .byte    $ff     ; 94
     A2A2   FF            7618           .byte    $ff     ; 95
     A2A3   FF            7619           .byte    $ff     ; 96
     A2A4   FF            7620           .byte    $ff     ; 97
                          7621
     A2A5   FF            7622           .byte    $ff     ; 98
     A2A6   FF            7623           .byte    $ff     ; 99
     A2A7   FF            7624           .byte    $ff     ; 9a
     A2A8   FF            7625           .byte    $ff     ; 9b
     A2A9   FF            7626           .byte    $ff     ; 9c
     A2AA   FF            7627           .byte    $ff     ; 9d
     A2AB   FF            7628           .byte    $ff     ; 9e
     A2AC   FF            7629           .byte    $ff     ; 9f
                          7630
     A2AD   FF            7631           .byte    $ff     ; a0
     A2AE   FF            7632           .byte    $ff     ; a1
     A2AF   FF            7633           .byte    $ff     ; a2
     A2B0   FF            7634           .byte    $ff     ; a3
     A2B1   08            7635           .byte    $08     ; a4
     A2B2   08            7636           .byte    $08     ; a5
     A2B3   08            7637           .byte    $08     ; a6
     A2B4   08            7638           .byte    $08     ; a7
                          7639
     A2B5   00            7640           .byte    $00     ; a8
     A2B6   00            7641           .byte    $00     ; a9
     A2B7   00            7642           .byte    $00     ; aa
     A2B8   00            7643           .byte    $00     ; ab
     A2B9   01            7644           .byte    $01     ; ac
     A2BA   01            7645           .byte    $01     ; ad
     A2BB   01            7646           .byte    $01     ; ae
     A2BC   01            7647           .byte    $01     ; af
                          7648
     A2BD   FF            7649           .byte    $ff     ; b0
     A2BE   FF            7650           .byte    $ff     ; b1
     A2BF   FF            7651           .byte    $ff     ; b2
     A2C0   FF            7652           .byte    $ff     ; b3
     A2C1   0C            7653           .byte    $0c     ; b4
     A2C2   0C            7654           .byte    $0c     ; b5
     A2C3   0C            7655           .byte    $0c     ; b6
     A2C4   0C            7656           .byte    $0c     ; b7
```

```
Error Addr  Code          Seq    Source statement

                          7657
      A2C5  04            7658            .byte   $04     ; b8
      A2C6  04            7659            .byte   $04     ; b9
      A2C7  04            7660            .byte   $04     ; ba
      A2C8  04            7661            .byte   $04     ; bb
      A2C9  05            7662            .byte   $05     ; bc
      A2CA  05            7663            .byte   $05     ; bd
      A2CB  05            7664            .byte   $05     ; be
      A2CC  05            7665            .byte   $05     ; bf
                          7666
      A2CD  FF            7667            .byte   $ff     ; c0
      A2CE  FF            7668            .byte   $ff     ; c1
      A2CF  FF            7669            .byte   $ff     ; c2
      A2D0  FF            7670            .byte   $ff     ; c3
      A2D1  FF            7671            .byte   $ff     ; c4
      A2D2  FF            7672            .byte   $ff     ; c5
      A2D3  FF            7673            .byte   $ff     ; c6
      A2D4  FF            7674            .byte   $ff     ; c7
                          7675
      A2D5  02            7676            .byte   $02     ; c8
      A2D6  02            7677            .byte   $02     ; c9
      A2D7  02            7678            .byte   $02     ; ca
      A2D8  02            7679            .byte   $02     ; cb
      A2D9  03            7680            .byte   $03     ; cc
      A2DA  03            7681            .byte   $03     ; cd
      A2DB  03            7682            .byte   $03     ; ce
      A2DC  03            7683            .byte   $03     ; cf
                          7684
      A2DD  FF            7685            .byte   $ff     ; d0
      A2DE  FF            7686            .byte   $ff     ; d1
      A2DF  FF            7687            .byte   $ff     ; d2
      A2E0  FF            7688            .byte   $ff     ; d3
      A2E1  0F            7689            .byte   $0f     ; d4
      A2E2  0F            7690            .byte   $0f     ; d5
      A2E3  0F            7691            .byte   $0f     ; d6
      A2E4  0F            7692            .byte   $0f     ; d7
                          7693
      A2E5  06            7694            .byte   $06     ; d8
      A2E6  06            7695            .byte   $06     ; d9
      A2E7  06            7696            .byte   $06     ; da
      A2E8  06            7697            .byte   $06     ; db
      A2E9  07            7698            .byte   $07     ; dc
      A2EA  07            7699            .byte   $07     ; dd
      A2EB  07            7700            .byte   $07     ; de
      A2EC  07            7701            .byte   $07     ; df
                          7702
      A2ED  FF            7703            .byte   $ff     ; e0
      A2EE  FF            7704            .byte   $ff     ; e1
      A2EF  FF            7705            .byte   $ff     ; e2
      A2F0  FF            7706            .byte   $ff     ; e3
```

```
Error Addr  Code           Seq    Source statement

     A2F1   09             7707            .byte    $09     ; e4
     A2F2   09             7708            .byte    $09     ; e5
     A2F3   09             7709            .byte    $09     ; e6
     A2F4   09             7710            .byte    $09     ; e7
                           7711
     A2F5   0A             7712            .byte    $0a     ; e8
     A2F6   0A             7713            .byte    $0a     ; e9
     A2F7   0A             7714            .byte    $0a     ; ea
     A2F8   0A             7715            .byte    $0a     ; eb
     A2F9   0B             7716            .byte    $0b     ; ec
     A2FA   0B             7717            .byte    $0b     ; ed
     A2FB   0B             7718            .byte    $0b     ; ee
     A2FC   0B             7719            .byte    $0b     ; ef
                           7720
     A2FD   FF             7721            .byte    $ff     ; f0
     A2FE   FF             7722            .byte    $ff     ; f1
     A2FF   FF             7723            .byte    $ff     ; f2
     A300   FF             7724            .byte    $ff     ; f3
     A301   0D             7725            .byte    $0d     ; f4
     A302   0D             7726            .byte    $0d     ; f5
     A303   0D             7727            .byte    $0d     ; f6
     A304   0D             7728            .byte    $0d     ; f7
                           7729
     A305   0E             7730            .byte    $0e     ; f8
     A306   0E             7731            .byte    $0e     ; f9
     A307   0E             7732            .byte    $0e     ; fa
     A308   0E             7733            .byte    $0e     ; fb
     A309   FF             7734            .byte    $ff     ; fc
     A30A   FF             7735            .byte    $ff     ; fd
     A30B   FF             7736            .byte    $ff     ; fe
     A30C   FF             7737            .byte    $ff     ; ff
                           7738            .include  gcrtable8
```

```
Error Addr  Code           Seq    Source statement

                           7740   ;       10/25/84 ds
                           7741
     A30D   FF             7742   gcrtb4  .byte    $ff     ; 0
     A30E   FF             7743           .byte    $ff     ; 1
     A30F   FF             7744           .byte    $ff     ; 2
     A310   FF             7745           .byte    $ff     ; 3
     A311   FF             7746           .byte    $ff     ; 4
     A312   FF             7747           .byte    $ff     ; 5
     A313   FF             7748           .byte    $ff     ; 6
     A314   FF             7749           .byte    $ff     ; 7
     A315   FF             7750           .byte    $ff     ; 8
     A316   08             7751           .byte    $08     ; 9
     A317   00             7752           .byte    $00     ; a
     A318   01             7753           .byte    $01     ; b
     A319   FF             7754           .byte    $ff     ; c
     A31A   0C             7755           .byte    $0c     ; d
     A31B   04             7756           .byte    $04     ; e
     A31C   05             7757           .byte    $05     ; f
     A31D   FF             7758           .byte    $ff     ; 10
     A31E   FF             7759           .byte    $ff     ; 11
     A31F   02             7760           .byte    $02     ; 12
     A320   03             7761           .byte    $03     ; 13
     A321   FF             7762           .byte    $ff     ; 14
     A322   0F             7763           .byte    $0f     ; 15
     A323   06             7764           .byte    $06     ; 16
     A324   07             7765           .byte    $07     ; 17
     A325   FF             7766           .byte    $ff     ; 18
     A326   09             7767           .byte    $09     ; 19
     A327   0A             7768           .byte    $0a     ; 1a
     A328   0B             7769           .byte    $0b     ; 1b
     A329   FF             7770           .byte    $ff     ; 1c
     A32A   0D             7771           .byte    $0d     ; 1d
     A32B   0E             7772           .byte    $0e     ; 1e
     A32C   FF             7773           .byte    $ff     ; 1f
     A32D   FF             7774           .byte    $ff     ; 20
     A32E   FF             7775           .byte    $ff     ; 21
     A32F   FF             7776           .byte    $ff     ; 22
     A330   FF             7777           .byte    $ff     ; 23
     A331   FF             7778           .byte    $ff     ; 24
     A332   FF             7779           .byte    $ff     ; 25
     A333   FF             7780           .byte    $ff     ; 26
     A334   FF             7781           .byte    $ff     ; 27
     A335   FF             7782           .byte    $ff     ; 28
     A336   08             7783           .byte    $08     ; 29
     A337   00             7784           .byte    $00     ; 2a
     A338   01             7785           .byte    $01     ; 2b
     A339   FF             7786           .byte    $ff     ; 2c
     A33A   0C             7787           .byte    $0c     ; 2d
     A33B   04             7788           .byte    $04     ; 2e
     A33C   05             7789           .byte    $05     ; 2f
```

```
Error Addr  Code           Seq    Source statement

     A33D   FF             7790            .byte   $ff     ; 30
     A33E   FF             7791            .byte   $ff     ; 31
     A33F   02             7792            .byte   $02     ; 32
     A340   03             7793            .byte   $03     ; 33
     A341   FF             7794            .byte   $ff     ; 34
     A342   0F             7795            .byte   $0f     ; 35
     A343   06             7796            .byte   $06     ; 36
     A344   07             7797            .byte   $07     ; 37
     A345   FF             7798            .byte   $ff     ; 38
     A346   09             7799            .byte   $09     ; 39
     A347   0A             7800            .byte   $0a     ; 3a
     A348   0B             7801            .byte   $0b     ; 3b
     A349   FF             7802            .byte   $ff     ; 3c
     A34A   0D             7803            .byte   $0d     ; 3d
     A34B   0E             7804            .byte   $0e     ; 3e
     A34C   FF             7805            .byte   $ff     ; 3f
     A34D   FF             7806            .byte   $ff     ; 40
     A34E   FF             7807            .byte   $ff     ; 41
     A34F   FF             7808            .byte   $ff     ; 42
     A350   FF             7809            .byte   $ff     ; 43
     A351   FF             7810            .byte   $ff     ; 44
     A352   FF             7811            .byte   $ff     ; 45
     A353   FF             7812            .byte   $ff     ; 46
     A354   FF             7813            .byte   $ff     ; 47
     A355   FF             7814            .byte   $ff     ; 48
     A356   08             7815            .byte   $08     ; 49
     A357   00             7816            .byte   $00     ; 4a
     A358   01             7817            .byte   $01     ; 4b
     A359   FF             7818            .byte   $ff     ; 4c
     A35A   0C             7819            .byte   $0c     ; 4d
     A35B   04             7820            .byte   $04     ; 4e
     A35C   05             7821            .byte   $05     ; 4f
                           7822
     A35D   FF             7823            .byte   $ff     ; 50
     A35E   FF             7824            .byte   $ff     ; 51
     A35F   02             7825            .byte   $02     ; 52
     A360   03             7826            .byte   $03     ; 53
     A361   FF             7827            .byte   $ff     ; 54
     A362   0F             7828            .byte   $0f     ; 55
     A363   06             7829            .byte   $06     ; 56
     A364   07             7830            .byte   $07     ; 57
                           7831
     A365   FF             7832            .byte   $ff     ; 58
     A366   09             7833            .byte   $09     ; 59
     A367   0A             7834            .byte   $0a     ; 5a
     A368   0B             7835            .byte   $0b     ; 5b
     A369   FF             7836            .byte   $ff     ; 5c
     A36A   0D             7837            .byte   $0d     ; 5d
     A36B   0E             7838            .byte   $0e     ; 5e
     A36C   FF             7839            .byte   $ff     ; 5f
```

```
Error Addr  Code           Seq    Source statement

                          7840
      A36D  FF            7841            .byte   $ff     ; 60
      A36E  FF            7842            .byte   $ff     ; 61
      A36F  FF            7843            .byte   $ff     ; 62
      A370  FF            7844            .byte   $ff     ; 63
      A371  FF            7845            .byte   $ff     ; 64
      A372  FF            7846            .byte   $ff     ; 65
      A373  FF            7847            .byte   $ff     ; 66
      A374  FF            7848            .byte   $ff     ; 67
                          7849
      A375  FF            7850            .byte   $ff     ; 68
      A376  08            7851            .byte   $08     ; 69
      A377  00            7852            .byte   $00     ; 6a
      A378  01            7853            .byte   $01     ; 6b
      A379  FF            7854            .byte   $ff     ; 6c
      A37A  0C            7855            .byte   $0c     ; 6d
      A37B  04            7856            .byte   $04     ; 6e
      A37C  05            7857            .byte   $05     ; 6f
                          7858
      A37D  FF            7859            .byte   $ff     ; 70
      A37E  FF            7860            .byte   $ff     ; 71
      A37F  02            7861            .byte   $02     ; 72
      A380  03            7862            .byte   $03     ; 73
      A381  FF            7863            .byte   $ff     ; 74
      A382  0F            7864            .byte   $0f     ; 75
      A383  06            7865            .byte   $06     ; 76
      A384  07            7866            .byte   $07     ; 77
                          7867
      A385  FF            7868            .byte   $ff     ; 78
      A386  09            7869            .byte   $09     ; 79
      A387  0A            7870            .byte   $0a     ; 7a
      A388  0B            7871            .byte   $0b     ; 7b
      A389  FF            7872            .byte   $ff     ; 7c
      A38A  0D            7873            .byte   $0d     ; 7d
      A38B  0E            7874            .byte   $0e     ; 7e
      A38C  FF            7875            .byte   $ff     ; 7f
                          7876
      A38D  FF            7877            .byte   $ff     ; 80
      A38E  FF            7878            .byte   $ff     ; 81
      A38F  FF            7879            .byte   $ff     ; 82
      A390  FF            7880            .byte   $ff     ; 83
      A391  FF            7881            .byte   $ff     ; 84
      A392  FF            7882            .byte   $ff     ; 85
      A393  FF            7883            .byte   $ff     ; 86
      A394  FF            7884            .byte   $ff     ; 87
                          7885
      A395  FF            7886            .byte   $ff     ; 88
      A396  08            7887            .byte   $08     ; 89
      A397  00            7888            .byte   $00     ; 8a
      A398  01            7889            .byte   $01     ; 8b
```

```
Error Addr  Code          Seq    Source statement

     A399   FF           7890            .byte   $ff     ; 8c
     A39A   0C           7891            .byte   $0c     ; 8d
     A39B   04           7892            .byte   $04     ; 8e
     A39C   05           7893            .byte   $05     ; 8f
                         7894
     A39D   FF           7895            .byte   $ff     ; 90
     A39E   FF           7896            .byte   $ff     ; 91
     A39F   02           7897            .byte   $02     ; 92
     A3A0   03           7898            .byte   $03     ; 93
     A3A1   FF           7899            .byte   $ff     ; 94
     A3A2   0F           7900            .byte   $0f     ; 95
     A3A3   06           7901            .byte   $06     ; 96
     A3A4   07           7902            .byte   $07     ; 97
                         7903
     A3A5   FF           7904            .byte   $ff     ; 98
     A3A6   09           7905            .byte   $09     ; 99
     A3A7   0A           7906            .byte   $0a     ; 9a
     A3A8   0B           7907            .byte   $0b     ; 9b
     A3A9   FF           7908            .byte   $ff     ; 9c
     A3AA   0D           7909            .byte   $0d     ; 9d
     A3AB   0E           7910            .byte   $0e     ; 9e
     A3AC   FF           7911            .byte   $ff     ; 9f
                         7912
     A3AD   FF           7913            .byte   $ff     ; a0
     A3AE   FF           7914            .byte   $ff     ; a1
     A3AF   FF           7915            .byte   $ff     ; a2
     A3B0   FF           7916            .byte   $ff     ; a3
     A3B1   FF           7917            .byte   $ff     ; a4
     A3B2   FF           7918            .byte   $ff     ; a5
     A3B3   FF           7919            .byte   $ff     ; a6
     A3B4   FF           7920            .byte   $ff     ; a7
                         7921
     A3B5   FF           7922            .byte   $ff     ; a8
     A3B6   08           7923            .byte   $08     ; a9
     A3B7   00           7924            .byte   $00     ; aa
     A3B8   01           7925            .byte   $01     ; ab
     A3B9   FF           7926            .byte   $ff     ; ac
     A3BA   0C           7927            .byte   $0c     ; ad
     A3BB   04           7928            .byte   $04     ; ae
     A3BC   05           7929            .byte   $05     ; af
                         7930
     A3BD   FF           7931            .byte   $ff     ; b0
     A3BE   FF           7932            .byte   $ff     ; b1
     A3BF   02           7933            .byte   $02     ; b2
     A3C0   03           7934            .byte   $03     ; b3
     A3C1   FF           7935            .byte   $ff     ; b4
     A3C2   0F           7936            .byte   $0f     ; b5
     A3C3   06           7937            .byte   $06     ; b6
     A3C4   07           7938            .byte   $07     ; b7
                         7939
```

```
Error Addr  Code          Seq   Source statement

     A3C5   FF           7940          .byte   $ff     ; b8
     A3C6   09           7941          .byte   $09     ; b9
     A3C7   0A           7942          .byte   $0a     ; ba
     A3C8   0B           7943          .byte   $0b     ; bb
     A3C9   FF           7944          .byte   $ff     ; bc
     A3CA   0D           7945          .byte   $0d     ; bd
     A3CB   0E           7946          .byte   $0e     ; be
     A3CC   FF           7947          .byte   $ff     ; bf
                         7948
     A3CD   FF           7949          .byte   $ff     ; c0
     A3CE   FF           7950          .byte   $ff     ; c1
     A3CF   FF           7951          .byte   $ff     ; c2
     A3D0   FF           7952          .byte   $ff     ; c3
     A3D1   FF           7953          .byte   $ff     ; c4
     A3D2   FF           7954          .byte   $ff     ; c5
     A3D3   FF           7955          .byte   $ff     ; c6
     A3D4   FF           7956          .byte   $ff     ; c7
                         7957
     A3D5   FF           7958          .byte   $ff     ; c8
     A3D6   08           7959          .byte   $08     ; c9
     A3D7   00           7960          .byte   $00     ; ca
     A3D8   01           7961          .byte   $01     ; cb
     A3D9   FF           7962          .byte   $ff     ; cc
     A3DA   0C           7963          .byte   $0c     ; cd
     A3DB   04           7964          .byte   $04     ; ce
     A3DC   05           7965          .byte   $05     ; cf
                         7966
     A3DD   FF           7967          .byte   $ff     ; d0
     A3DE   FF           7968          .byte   $ff     ; d1
     A3DF   02           7969          .byte   $02     ; d2
     A3E0   03           7970          .byte   $03     ; d3
     A3E1   FF           7971          .byte   $ff     ; d4
     A3E2   0F           7972          .byte   $0f     ; d5
     A3E3   06           7973          .byte   $06     ; d6
     A3E4   07           7974          .byte   $07     ; d7
                         7975
     A3E5   FF           7976          .byte   $ff     ; d8
     A3E6   09           7977          .byte   $09     ; d9
     A3E7   0A           7978          .byte   $0a     ; da
     A3E8   0B           7979          .byte   $0b     ; db
     A3E9   FF           7980          .byte   $ff     ; dc
     A3EA   0D           7981          .byte   $0d     ; dd
     A3EB   0E           7982          .byte   $0e     ; de
     A3EC   FF           7983          .byte   $ff     ; df
                         7984
     A3ED   FF           7985          .byte   $ff     ; e0
     A3EE   FF           7986          .byte   $ff     ; e1
     A3EF   FF           7987          .byte   $ff     ; e2
     A3F0   FF           7988          .byte   $ff     ; e3
     A3F1   FF           7989          .byte   $ff     ; e4
```

```
Error Addr  Code           Seq    Source statement

     A3F2   FF             7990           .byte    $ff     ; e5
     A3F3   FF             7991           .byte    $ff     ; e6
     A3F4   FF             7992           .byte    $ff     ; e7
                           7993
     A3F5   FF             7994           .byte    $ff     ; e8
     A3F6   08             7995           .byte    $08     ; e9
     A3F7   00             7996           .byte    $00     ; ea
     A3F8   01             7997           .byte    $01     ; eb
     A3F9   FF             7998           .byte    $ff     ; ec
     A3FA   0C             7999           .byte    $0c     ; ed
     A3FB   04             8000           .byte    $04     ; ee
     A3FC   05             8001           .byte    $05     ; ef
                           8002
     A3FD   FF             8003           .byte    $ff     ; f0
     A3FE   FF             8004           .byte    $ff     ; f1
     A3FF   02             8005           .byte    $02     ; f2
     A400   03             8006           .byte    $03     ; f3
     A401   FF             8007           .byte    $ff     ; f4
     A402   0F             8008           .byte    $0f     ; f5
     A403   06             8009           .byte    $06     ; f6
     A404   07             8010           .byte    $07     ; f7
                           8011
     A405   FF             8012           .byte    $ff     ; f8
     A406   09             8013           .byte    $09     ; f9
     A407   0A             8014           .byte    $0a     ; fa
     A408   0B             8015           .byte    $0b     ; fb
     A409   FF             8016           .byte    $ff     ; fc
     A40A   0D             8017           .byte    $0d     ; fd
     A40B   0E             8018           .byte    $0e     ; fe
     A40C   FF             8019           .byte    $ff     ; ff
                           8020           .include  duplct1
```

```
Error Addr  Code           Seq    Source statement

                           8022            .subttl 'duplct1.src'
                           8023
                           8024    ;  controller format disk
                           8025
     A40D   A9 47          8026    jformat lda  #71
     A40F   8D 02AC        8027            sta  maxtrk     ; double sided
     A412   A9 03          8028            lda  #3
     A414   20 D6D3        8029            jsr  seth
     A417   A2 03          8030            ldx  #3         ; job #3
     A419   A9 00          8031            lda  #0
     A41B   8D 01B2        8032            sta  fmtsid     ; side zero first
     A41E   A9 F0          8033            lda  #$f0       ; format cmd
     A420   85 3B          8034            sta  tsttrk     ; init speed var
     A422   95 00          8035            sta  jobs,x     ; give job to controller
     A424   20 9FB6        8036            jsr  stbctl     ; wake him up
     A427   C9 02          8037            cmp  #2         ; error?
     A429   B0 45          8038            bcs  1$         ; br, error
                           8039
                           8040    ;read track one sector zero
                           8041
     A42B   A0 03          8042            ldy  #3         ; retries
     A42D   A9 01          8043    4$      lda  #1         ; track 1
     A42F   85 0C          8044            sta  hdrs+6     ; *
     A431   A9 00          8045            lda  #0         ; sector 0
     A433   85 0D          8046            sta  hdrs+7     ; *
     A435   A9 80          8047            lda  #$80       ; read
     A437   95 00          8048            sta  jobs,x     ; give job to controller
     A439   20 9FB6        8049            jsr  stbctl     ; wake him up
     A43C   C9 02          8050            cmp  #2         ; error?
     A43E   90 05          8051            bcc  5$         ; br, ok...
                           8052
     A440   88             8053            dey
     A441   10 EA          8054            bpl  4$         ; try 3 times
     A443   B0 2B          8055            bcs  1$         ; bra
                           8056
     A445   A9 01          8057    5$      lda  #1
     A447   8D 01B2        8058            sta  fmtsid     ; side one second
     A44A   A9 F0          8059            lda  #$f0       ; format cmd
     A44C   85 3B          8060            sta  tsttrk     ; init speed var
     A44E   95 00          8061            sta  jobs,x     ; give job to controller
     A450   20 9FB6        8062            jsr  stbctl     ; wake him up
     A453   C9 02          8063            cmp  #2         ; error?
     A455   B0 19          8064            bcs  1$         ; br, error
                           8065
                           8066    ;read track thirty-six sector zero
                           8067
     A457   A0 03          8068            ldy  #3         ; retries
     A459   A9 24          8069    6$      lda  #36        ; track 36
     A45B   85 0C          8070            sta  hdrs+6     ; *
     A45D   A9 00          8071            lda  #0         ; sector 0
```

```
Error Addr  Code           Seq    Source statement

     A45F   85 0D         8072            sta  hdrs+7      ; *
     A461   A9 80         8073            lda  #$80        ; read
     A463   95 00         8074            sta  jobs,x      ; give job to controller
     A465   20 9FB6       8075            jsr  stbctl      ; wake him up
     A468   C9 02         8076            cmp  #2          ; error?
     A46A   B0 01         8077            bcs  3$          ; br, bad
                          8078
     A46C   60            8079            rts              ; ok
                          8080
     A46D   88            8081    3$      dey
     A46E   10 E9         8082            bpl  6$          ; keep trying
                          8083
     A470   A2 00         8084    1$      ldx  #0          ; set for offset for buffer to det. trk & sect.
     A472   2C 0298       8085            bit  jobrtn      ; return on error ?
     A475   8E 0298       8086            stx  jobrtn      ; clr
     A478   10 01         8087            bpl  7$
                          8088
     A47A   60            8089            rts              ; back to caller
     A47B   4C E60A       8090    7$      jmp  error
                          8091            .include  patch
```

```
Error Addr  Code           Seq    Source statement

                           8093            .subttl 'patch.src'
                           8094
                           8095    ;*****************************************************
                           8096
     A47E   8A             8097    jslower txa
     A47F   A2 05          8098            ldx  #5
     A481   D0 03          8099            bne  jslowd+3   ; bra
                           8100
     A483   8A             8101    jslowd  txa
     A484   A2 0D          8102            ldx  #$0d       ; insert 40 us. delay at 2 Mhz
     A486   CA             8103    1$      dex
     A487   D0 FD          8104            bne  1$
     A489   AA             8105            tax
     A48A   60             8106            rts
                           8107
                           8108    ;*****************************************************
                           8109
     A48B   A5 6D          8110    sav_pnt lda  bmpnt      ; save pointers
     A48D   8D 02AD        8111            sta  savbm
     A490   A5 6E          8112            lda  bmpnt+1
     A492   8D 02AE        8113            sta  savbm+1
     A495   60             8114            rts
                           8115
                           8116    ;*****************************************************
                           8117
     A496   AD 02AD        8118    res_pnt lda  savbm
     A499   85 6D          8119            sta  bmpnt      ; save pointers
     A49B   AD 02AE        8120            lda  savbm+1
     A49E   85 6E          8121            sta  bmpnt+1
     A4A0   60             8122            rts
                           8123
                           8124    ;*****************************************************
                           8125
     A4A1   A6 7F          8126    set_bm  ldx  drvnum
                           8127
     A4A3                  8128+           NODRRD          ; read nodrv,x absolute
     A4A3   BD FF 00       8129A           .byte $bd,$ff,$00
                           8130    ;       lda  nodrv,x    ; drive active
                           8131
     A4A6   F0 05          8132            beq  1$
                           8133
     A4A8   A9 74          8134            lda  #nodriv
     A4AA   20 E648        8135            jsr  cmder3
                           8136
     A4AD   20 F119        8137    1$      jsr  bam2x      ; get index into bufx
     A4B0   20 F0DF        8138            jsr  redbam     ; read BAM if neccessary
     A4B3   AD 02F9        8139            lda  wbam       ; write pending ?
     A4B6   F0 07          8140            beq  3$
                           8141
     A4B8   09 80          8142            ora  #$80
```

```
Error Addr  Code           Seq    Source statement

     A4BA   8D 02F9       8143            sta  wbam        ; set write pending flag
     A4BD   D0 03         8144            bne  2$          ; bra*
                          8145
     A4BF   20 A58D       8146    3$      jsr  wrt_bam     ; write out BAM
                          8147
     A4C2   20 A48B       8148    2$      jsr  sav_pnt     ; save current BAM pointers
     A4C5   20 A534       8149            jsr  where       ; find BAM
     A4C8   A5 80         8150            lda  track       ; get offset
     A4CA   38            8151            sec
     A4CB   E9 24         8152            sbc  #36
     A4CD   A8            8153            tay
     A4CE   B1 6D         8154            lda  (bmpnt),y   ; get count
     A4D0   48            8155            pha
     A4D1   20 A496       8156            jsr  res_pnt     ; restore BAM pointers
     A4D4   68            8157            pla              ; count in .a
     A4D5   60            8158            rts
                          8159
                          8160    ;*****************************************************
            =A4E7         8161            *=*+17           ; adjust address
                          8162    ;*****************************************************
                          8163
     A4E7   A5 80         8164    bam_pt  lda  track
     A4E9   38            8165            sec
     A4EA   E9 24         8166            sbc  #36         ; offset
     A4EC   A8            8167            tay
     A4ED   A5 81         8168            lda  sector      ; sector/8
     A4EF   4A            8169            lsr  a
     A4F0   4A            8170            lsr  a
     A4F1   4A            8171            lsr  a
     A4F2   18            8172            clc
     A4F3   79 A5DB       8173            adc  bmindx,y    ; get location
     A4F6   A8            8174            tay
     A4F7   A5 81         8175            lda  sector
     A4F9   29 07         8176            and  #7
     A4FB   AA            8177            tax              ; which sector
     A4FC   B9 0146       8178            lda  ovrbuf+$46,y
     A4FF   3D EFE9       8179            and  bmask,x
     A502   08            8180            php              ; save status
     A503   B9 0146       8181            lda  ovrbuf+$46,y
     A506   28            8182            plp              ; retrieve status set/clr (z)
     A507   60            8183            rts
                          8184
                          8185    ;*****************************************************
                          8186
     A508   20 A48B       8187    deall_b jsr  sav_pnt     ; save pointers
     A50B   20 A534       8188            jsr  where       ; where is the BAM
     A50E   A5 80         8189            lda  track
     A510   38            8190            sec
     A511   E9 24         8191            sbc  #36         ; offset
     A513   A8            8192            tay.             ; index into table
```

```
Error Addr  Code           Seq    Source statement

     A514   18             8193            clc
     A515   B1 6D          8194            lda  (bmpnt),y  ; goto location and add 1
     A517   69 01          8195            adc  #1
     A519   91 6D          8196            sta  (bmpnt),y
     A51B   4C A496        8197            jmp  res_pnt    ; restore BAM pointers
                           8198
                           8199    ;*****************************************************
                           8200
     A51E   20 A48B        8201    alloc_b jsr  sav_pnt    ; save pointers
     A521   20 A534        8202            jsr  where      ; find location of the BAM
     A524   A5 80          8203            lda  track
     A526   38             8204            sec
     A527   E9 24          8205            sbc  #36        ; offset
     A529   A8             8206            tay             ; index into table
     A52A   38             8207            sec
     A52B   B1 6D          8208            lda  (bmpnt),y  ; goto location and sub 1
     A52D   E9 01          8209            sbc  #1
     A52F   91 6D          8210            sta  (bmpnt),y
     A531   4C A496        8211            jmp  res_pnt    ; restore pointers
                           8212
                           8213    ;*****************************************************
                           8214
     A534   A2 0D          8215    where   ldx  #blindx+7
     A536   B5 A7          8216            lda  buf0,x
     A538   29 0F          8217            and  #$0f
     A53A   AA             8218            tax
     A53B   BD FEE0        8219            lda  bufind,x   ; which buffer is the BAM in
     A53E   85 6E          8220            sta  bmpnt+1
     A540   A9 DD          8221            lda  #$dd
     A542   85 6D          8222            sta  bmpnt
     A544   60             8223            rts
                           8224
                           8225    ;*****************************************************
                           8226
     A545   A5 6F          8227    chk_blk lda  temp
     A547   48             8228            pha             ; save temp
     A548   A5 80          8229            lda  track
     A54A   38             8230            sec
     A54B   E9 24          8231            sbc  #36        ; get offset
     A54D   A8             8232            tay
     A54E   48             8233            pha             ; save
     A54F   20 A48B        8234            jsr  sav_pnt
     A552   20 A534        8235            jsr  where      ; where is the BAM
     A555   B1 6D          8236            lda  (bmpnt),y
     A557   48             8237            pha
     A558   A9 00          8238            lda  #0
     A55A   85 6F          8239            sta  temp       ; start at zero
     A55C   A9 01          8240            lda  #>bam_one
     A55E   85 6E          8241            sta  bmpnt+1    ; msb
     A560   B9 A5DB        8242            lda  bmindx,y   ; starts here
```

```
Error Addr  Code           Seq    Source statement

     A563   18             8243            clc
     A564   69 46          8244            adc  #<bam_one  ; add offset
     A566   85 6D          8245            sta  bmpnt
                           8246
     A568   A0 02          8247            ldy  #2
     A56A   A2 07          8248   1$       ldx  #7         ; verify bit map to count value
     A56C   B1 6D          8249   2$       lda  (bmpnt),y
     A56E   3D EFE9        8250            and  bmask,x
     A571   F0 02          8251            beq  3$
                           8252
     A573   E6 6F          8253            inc  temp       ; on free increment
     A575   CA             8254   3$       dex
     A576   10 F4          8255            bpl  2$
                           8256             .
     A578   88             8257            dey
     A579   10 EF          8258            bpl  1$
                           8259
     A57B   68             8260            pla
     A57C   C5 6F          8261            cmp  temp
     A57E   F0 05          8262            beq  4$
                           8263
     A580   A9 71          8264            lda  #direrr
     A582   20 E645        8265            jsr  cmder2
                           8266
     A585   68             8267   4$       pla
     A586   A8             8268            tay             ; restore track offset
     A587   68             8269            pla
     A588   85 6F          8270            sta  temp       ; restore temp
     A58A   4C A496        8271            jmp  res_pnt
                           8272
                           8273   ;*************************************************
                           8274
     A58D   AD 180F        8275   wrt_bam lda  pota1      ; 1/2 Mhz ?
     A590   29 20          8276            and  #$20
     A592   D0 03          8277            bne  1$
                           8278
     A594   4C D58A        8279   2$       jmp  dowrit
                           8280
     A597   AD 02AC        8281   1$       lda  maxtrk
     A59A   C9 25          8282            cmp  #37
     A59C   90 F6          8283            bcc  2$
                           8284
     A59E   A6 F9          8285            ldx  jobnum
     A5A0   BD 025B        8286            lda  lstjob,x   ; save last job
     A5A3   48             8287            pha
     A5A4   20 D58A        8288            jsr  dowrit     ; write out side zero
     A5A7   20 A48B        8289            jsr  sav_pnt    ; save bmpnt
     A5AA   20 EF3A        8290            jsr  setbpt
     A5AD   20 F008        8291            jsr  clrbam+3   ; clear buffer
     A5B0   A5 F9          8292            lda  jobnum
```

```
Error Addr  Code           Seq    Source statement

     A5B2   0A            8293            asl  a
     A5B3   AA            8294            tax
     A5B4   A9 35         8295            lda  #53
     A5B6   95 06         8296            sta  hdrs,x      ; put track in job queue, same sector
                          8297
     A5B8   A0 68         8298            ldy  #104
     A5BA   B9 0146       8299    3$      lda  ovrbuf+$46,y
     A5BD   91 6D         8300            sta  (bmpnt),y
     A5BF   88            8301            dey
     A5C0   10 F8         8302            bpl  3$          ; transfer to buffer
                          8303
     A5C2   20 A496       8304            jsr  res_pnt     ; restore pointers
     A5C5   20 D58A       8305            jsr  dowrit      ; write this track out
                          8306
     A5C8   A5 F9         8307            lda  jobnum
     A5CA   0A            8308            asl  a
     A5CB   AA            8309            tax
     A5CC   AD FE85       8310            lda  dirtrk      ; *2
     A5CF   95 06         8311            sta  hdrs,x      ; read back BAM side zero
     A5D1   20 D586       8312            jsr  doread      ; done
     A5D4   68            8313            pla
     A5D5   A6 F9         8314            ldx  jobnum
     A5D7   9D 025B       8315            sta  lstjob,x    ; restore last job
     A5DA   60            8316            rts
                          8317    ;*****************************************************
                          8318
     A5DB   00 03 06      8319    bmindx  .byte 0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54
     A5DE   09 0C 0F
     A5E1   12 15 18
     A5E4   1B 1E 21
     A5E7   24 27 2A
     A5EA   2D 30 33
     A5ED   36
     A5EE   39 3C 3F      8320            .byte 57,60,63,66,69,72,75,78,81,84,87,90,93,96,99,102
     A5F1   42 45 48
     A5F4   4B 4E 51
     A5F7   54 57 5A
     A5FA   5D 60 63
     A5FD   66
                          8321
```

```
Error Addr  Code           Seq   Source statement

                          8323   ;-------------------------------------------------
                          8324   ;
                          8325   ; PATCHES  David G Siracusa - Commodore (c) 1985
                          8326   ;
                          8327   ; Compatibility is fun...
                          8328   ; I know you can say fun... I know you can...
                          8329   ;
                          8330   ;-------------------------------------------------
                          8331   ;
                          8332   ;     patch 18  *rom ds 01/21/85*
                          8333   ;
     A5FE   AD 180F       8334   ptch18  lda  pota1      ; 1/2 Mhz ?
     A601   29 20         8335           and  #$20
     A603   F0 0F         8336           beq  1$
                          8337
     A605   A0 00         8338           ldy  #0          ; clr regs
     A607   A2 00         8339           ldx  #0
     A609   A9 01         8340           lda  #1          ; place filename
     A60B   8D 027A       8341           sta  filtbl
     A60E   20 C312       8342           jsr  onedrv      ; setup drv
     A611   4C E7A8       8343           jmp  rtch18      ; ret
                          8344
     A614   A9 8D         8345   1$      lda  #$8d        ; 1541 mode
     A616   20 C268       8346           jsr  parse
     A619   4C E7A8       8347           jmp  rtch18      ; ret
                          8348   ;
                          8349   ;
                          8350   ;-------------------------------------------------
                          8351   ;
                          8352   ;     patch 19  *rom ds 01/21/85*
                          8353   ;
     A61C   20 C146       8354   ptch19  jsr  parsxq     ; parse & xeq cmd
     A61F   20 81B2       8355           jsr  spinp       ; input
     A622   A5 37         8356           lda  fastsr      ; clr error
     A624   29 7F         8357           and  #$7f
     A626   85 37         8358           sta  fastsr
     A628   4C EBFF       8359           jmp  rtch19
                          8360   ;
                          8361   ;
                          8362   ;-------------------------------------------------
                          8363   ;
                          8364   ;     patch 20  *rom ds 01/21/85*
                          8365   ;
     A62B   A9 FF         8366   ptch20  lda  #255
     A62D   85 48         8367           sta  acltim
     A62F   A9 06         8368           lda  #6          ; setup timer2
     A631   85 35         8369           sta  acltim2
     A633   60            8370           rts
                          8371   ;
                          8372   ;
```

```
Error Addr  Code           Seq    Source statement

                           8373   ;-------------------------------------------------
                           8374   ;
                           8375   ;     patch 21  *rom ds 01/21/85*
                           8376   ;
     A634   D0 07          8377   ptch21  bne  1$
                           8378
     A636   AD 02AB        8379           lda  mtrcnt      ; anything to do ?
     A639   D0 10          8380           bne  2$
     A63B   F0 1A          8381           beq  3$
                           8382
     A63D   A9 FF          8383   1$      lda  #255
     A63F   8D 02AB        8384           sta  mtrcnt      ; irq * 255
     A642   20 8764        8385           jsr  moton       ; turn on the motor
     A645   A9 01          8386           lda  #1
     A647   85 1C          8387           sta  wpsw
     A649   D0 0C          8388           bne  3$          ; bra
                           8389
     A64B   CE 02AB        8390   2$      dec  mtrcnt
     A64E   D0 07          8391           bne  3$
                           8392
     A650   A5 20          8393           lda  drvst
     A652   D0 03          8394           bne  3$
                           8395
     A654   20 8770        8396           jsr  motoff      ; turn off the motor
                           8397
     A657   4C F9B1        8398   3$      jmp  rtch21      ; return
                           8399   ;
                           8400   ;
                           8401   ;-------------------------------------------------
                           8402   ;
                           8403   ;     patch 22
                           8404   ;
     A65A   A9 02          8405   ptch22  lda  #2
     A65C   8D 1800        8406           sta  pb
     A65F   A9 20          8407           lda  #$20        ; *,_atn,2 mhz,*,*,side 0,sr-in,*
     A661   8D 1801        8408           sta  pa1
     A664   4C FF18        8409           jmp  ptch22r     ; ret
                           8410   ;
                           8411   ;
                           8412   ;-------------------------------------------------
                           8413   ;
                           8414   ;     patch 23
                           8415   ;
     A667   AD 180F        8416   ptch23  lda  pota1      ; 1/2 Mhz ?
     A66A   29 20          8417           and  #$20
     A66C   D0 03          8418           bne  1$
                           8419
     A66E   4C D586        8420   2$      jmp  doread      ; 1541 mode read BAM side 1
                           8421
     A671   AD 02AC        8422   1$      lda  maxtrk      ; seek ok, on other side ?
```

```
Error Addr  Code           Seq    Source statement

     A674   C9 25         8423           cmp  #37
     A676   90 F6         8424           bcc  2$          ; seek regular method
                          8425
     A678   20 A48B       8426           jsr  sav_pnt     ; save pointers
                          8427
     A67B   A9 00         8428           lda  #0
     A67D   85 6D         8429           sta  bmpnt       ; even page
     A67F   A6 F9         8430           ldx  jobnum
     A681   BD FEE0       8431           lda  bufind,x    ; which buffer is it in
     A684   85 6E         8432           sta  bmpnt+1
     A686   A9 FF         8433           lda  #$ff
     A688   8D 0298       8434           sta  jobrtn      ; error recovery on
     A68B   A5 F9         8435           lda  jobnum
     A68D   0A            8436           asl  a
     A68E   AA            8437           tax
     A68F   A9 35         8438           lda  #53         ; read BAM side one
     A691   95 06         8439           sta  hdrs,x      ; put track in job queue, same sector
     A693   20 D586       8440           jsr  doread      ; read it
     A696   C9 02         8441           cmp  #2          ; ok ?
     A698   6A            8442           ror  a           ; carry set bad return
     A699   29 80         8443           and  #$80        ; isolate sign bit
     A69B   49 80         8444           eor  #$80
     A69D   8D 01AF       8445           sta  dside
     A6A0   10 0A         8446           bpl  3$          ; br, error
                          8447
     A6A2   A0 68         8448           ldy  #104
     A6A4   B1 6D         8449   4$      lda  (bmpnt),y   ; get BAM 1 and put somewhere in God's country
     A6A6   99 0146       8450           sta  ovrbuf+$46,y
     A6A9   88            8451           dey
     A6AA   10 F8         8452           bpl  4$
                          8453
     A6AC   A9 FF         8454   3$      lda  #$ff
     A6AE   8D 0298       8455           sta  jobrtn      ; set error recovery on
     A6B1   A5 F9         8456           lda  jobnum
     A6B3   0A            8457           asl  a
     A6B4   AA            8458           tax
     A6B5   AD FE85       8459           lda  dirtrk      ; read BAM side zero
     A6B8   95 06         8460           sta  hdrs,x      ; put track in job queue, same sector
     A6BA   20 D586       8461           jsr  doread      ; read it
     A6BD   C9 02         8462           cmp  #2
     A6BF   90 10         8463           bcc  5$
                          8464
     A6C1   AA            8465           tax              ; save error
     A6C2   A9 24         8466           lda  #36
     A6C4   8D 02AC       8467           sta  maxtrk      ; def single sided
     A6C7   20 A496       8468           jsr  res_pnt     ; save BAM pointers
     A6CA   8A            8469           txa
     A6CB   20 E60A       8470           jsr  error       ; let the error handler do the rest
     A6CE   4C D644       8471           jmp  rec7        ; consist
                          8472
```

```
Error Addr  Code           Seq    Source statement

     A6D1   A0 03         8473   5$       ldy  #3
     A6D3   B1 6D         8474            lda  (bmpnt),y  ; check double sided flag
     A6D5   2D 01AF       8475            and  dside      ; both must be ok....
     A6D8   30 03         8476            bmi  6$
                          8477
     A6DA   A9 24         8478            lda  #36
     A6DC   2C            8479            .byte skip2
     A6DD   A9 47         8480   6$       lda  #71
     A6DF   8D 02AC       8481            sta  maxtrk     ; double sided
     A6E2   4C A496       8482            jmp  res_pnt    ; restore BAM pointers
                          8483   ;
                          8484   ;
                          8485   ;-------------------------------------------------
                          8486   ;
                          8487   ;     patch 24
                          8488   ;
     A6E5   20 D58C       8489   ptch24   jsr  dojob      ; seek side zero
     A6E8   48            8490            pha             ; save status
     A6E9   C9 02         8491            cmp  #2         ; error ?
     A6EB   B0 49         8492            bcs  1$         ; br, error...
                          8493
     A6ED   AD 180F       8494            lda  pota1
     A6F0   29 20         8495            and  #$20
     A6F2   F0 42         8496            beq  1$         ; 1/2 Mhz ?
                          8497
     A6F4   A9 47         8498            lda  #71
     A6F6   8D 02AC       8499            sta  maxtrk     ; let DOS think he has double sided capability
                          8500
     A6F9   A9 FF         8501            lda  #$ff
     A6FB   8D 0298       8502            sta  jobrtn     ; return from error
     A6FE   A5 16         8503            lda  header     ; get id's
     A700   48            8504            pha
     A701   A5 17         8505            lda  header+1
     A703   48            8506            pha             ; save them
                          8507
     A704   A5 F9         8508            lda  jobnum
     A706   0A            8509            asl  a
     A707   AA            8510            tax
     A708   A9 35         8511            lda  #53        ; seek side one BAM track
     A70A   95 06         8512            sta  hdrs,x     ; put track in job queue, same sector
     A70C   A9 B0         8513            lda  #seek
     A70E   20 D58C       8514            jsr  dojob      ; try it...
     A711   C9 02         8515            cmp  #2         ; error ?
                          8516
     A713   68            8517            pla
     A714   A8            8518            tay             ; header+1
     A715   68            8519            pla
     A716   AA            8520            tax             ; header
     A717   B0 0B         8521            bcs  2$         ; error on last job ?
                          8522
```

```
Error Addr  Code          Seq   Source statement

     A719   E4 16        8523           cpx  header      ; same id's ?
     A71B   DO 07        8524           bne  2$
                         8525
     A71D   C4 17        8526           cpy  header+1    ; same id's ?
     A71F   DO 03        8527           bne  2$
                         8528
     A721   A9 47        8529           lda  #71         ; probably double sided
     A723   2C           8530           .byte skip2
     A724   A9 24        8531   2$      lda  #36
     A726   8D O2AC      8532           sta  maxtrk
                         8533
     A729   84 17        8534           sty  header+1    ; restore original id's from side zero
     A72B   86 16        8535           stx  header
                         8536
     A72D   A5 F9        8537           lda  jobnum
     A72F   OA           8538           asl  a
     A73O   AA           8539           tax
     A731   AD FE85      8540           lda  dirtrk
     A734   95 O6        8541           sta  hdrs,x      ; put track in job queue, same sector
                         8542
     A736   68           8543   1$      pla              ; get status
     A737   6O           8544           rts
                         8545   ;
                         8546   ;
                         8547   ;-----------------------------------------------------
                         8548   ;
                         8549   ;     patch 25
                         8550   ;
     A738   2O EF3A      8551   ptch25  jsr  setbpt      ; setup bit map pointer
     A73B   AD 18OF      8552           lda  pota1
     A73E   29 2O        8553           and  #%00100000 ; 1/2 Mhz ?
     A74O   FO OA        8554           beq  1$
                         8555
     A742   A9 OO        8556           lda  #0
     A744   AO 68        8557           ldy  #104        ; clr BAM side one
     A746   99 O146      8558   2$      sta  ovrbuf+$46,y
     A749   88           8559           dey
     A74A   1O FA        8560           bpl  2$
                         8561
     A74C   4C FOO8      8562   1$      jmp  rtch25      ; return
                         8563   ;
                         8564   ;
                         8565   ;-----------------------------------------------------
                         8566   ;
                         8567   ;     patch 26
                         8568   ;
     A74F   48           8569   ptch26  pha              ; save .a
     A75O   AD 18OF      8570           lda  pota1
     A753   29 2O        8571           and  #%00100000 ; 1/2 Mhz ?
     A755   FO O8        8572           beq  1$
```

```
Error Addr  Code           Seq    Source statement

                           8573
      A757  68             8574            pla
      A758  C9 24          8575            cmp  #36        ; > trk 35 ?
      A75A  90 04          8576            bcc  2$
                           8577
      A75C  E9 23          8578            sbc  #35
      A75E  24             8579            .byte skip1
      A75F  68             8580   1$       pla             ; restore .a
      A760  AE FED6        8581   2$       ldx  nzones     ; cont
      A763  60             8582            rts
                           8583   ;
                           8584   ;
                           8585   ;-------------------------------------------------
                           8586   ;
                           8587   ;     patch 27
                           8588   ;
      A764  20 F005        8589   ptch27   jsr  clrbam     ; clear BAM area
      A767  AD 180F        8590            lda  pota1
      A76A  29 20          8591            and  #%00100000
      A76C  D0 03          8592            bne  1$         ; 1/2 Mhz ?
                           8593
      A76E  A9 24          8594            lda  #36
      A770  2C             8595            .byte skip2
      A771  A9 47          8596   1$       lda  #71
      A773  8D 02AC        8597            sta  maxtrk     ; set maximum track full format
      A776  4C EE43        8598            jmp  rtch27     ; return
                           8599   ;
                           8600   ;
                           8601   ;-------------------------------------------------
                           8602   ;
                           8603   ;     patch 28
                           8604   ;
      A779  AD 180F        8605   ptch28   lda  pota1
      A77C  29 20          8606            and  #%00100000
      A77E  D0 03          8607            bne  1$         ; 1/2 Mhz ?
                           8608
      A780  4C C8C6        8609            jmp  format     ; 1541 mode
                           8610
      A783  4C A40D        8611   1$       jmp  jformat    ; 1571 mode
                           8612   ;
                           8613   ;
                           8614   ;-------------------------------------------------
                           8615   ;
                           8616   ;     patch 29
                           8617   ;
      A786  AD 1801        8618   ptch29   lda  pa1       ; change to 1 mhz
      A789  29 DF          8619            and  #%11011111
      A78B  8D 1801        8620            sta  pa1
      A78E  20 A483        8621            jsr  jslowd     ; wait for ...
                           8622
```

```
Error Addr  Code           Seq    Source statement

     A791   A9 7F         8623            lda  #$7f
     A793   8D 400D       8624            sta  icr         ; clear all sources of irq's
     A796   A9 08         8625            lda  #8
     A798   8D 400E       8626            sta  cra         ; turn off timers
     A79B   8D 400F       8627            sta  cra+1       ; off him too...
                          8628
     A79E   A9 00         8629            lda  #0          ; msb nil
     A7A0   8D 4005       8630            sta  tima_h
     A7A3   A9 06         8631            lda  #6          ; msb nil
     A7A5   8D 4004       8632            sta  tima_l      ; 6 us bit cell time
     A7A8   A9 01         8633            lda  #1
     A7AA   8D 400E       8634            sta  cra         ; start timer a
     A7AD   20 81B2       8635            jsr  spinp       ; finish up and enable irq's from '26
     A7B0   4C EA59       8636            jmp  tstatn      ; chk for atn & goto xidle
                          8637    ;
                          8638    ;
                          8639    ;-------------------------------------------------
                          8640    ;
                          8641    ;     patch 30
                          8642    ;
     A7B3   AD 180F       8643    ptch30  lda  pota1
     A7B6   29 20         8644            and  #%00100000
     A7B8   F0 03         8645            beq  1$          ; 1/2 Mhz ?
                          8646
     A7BA   4C 80CE       8647            jmp  jatnsrv     ; 2 Mhz
     A7BD   4C E85B       8648    1$      jmp  atnsrv      ; 1 Mhz
                          8649    ;
                          8650    ;
                          8651    ;-------------------------------------------------
                          8652    ;
                          8653    ;     patch 31
                          8654    ;
     A7C0   78            8655    ptch31  sei
     A7C1   A2 45         8656            ldx  #topwrt     ; set stack pointer
     A7C3   9A            8657            txs
     A7C4   4C EB25       8658            jmp  rtch31
                          8659    ;
                          8660    ;
                          8661    ;-------------------------------------------------
                          8662    ;
                          8663    ;     patch 32
                          8664    ;
     A7C7   AD 180F       8665    ptch32  lda  pota1       ; 1/2 Mhz ?
     A7CA   29 20         8666            and  #$20
     A7CC   D0 09         8667            bne  1$
                          8668
     A7CE   A0 03         8669    6$      ldy  #3
     A7D0   A9 00         8670            lda  #0
     A7D2   91 6D         8671            sta  (bmpnt),y   ; set single sided bit in BAM side 0
     A7D4   4C EEB7       8672            jmp  newmap      ; finish up 1541 mode
```

```
Error Addr  Code           Seq    Source statement

                           8673
      A7D7  AD 02AC        8674  1$      lda  maxtrk      ; single/double sided?
      A7DA  C9 25          8675          cmp  #37
      A7DC  90 F0          8676          bcc  6$
                           8677
      A7DE  A0 01          8678          ldy  #1
      A7E0  A2 00          8679          ldx  #0
      A7E2  C0 12          8680  2$      cpy  #18
      A7E4  F0 34          8681          beq  4$
                           8682
      A7E6  8A             8683          txa
      A7E7  48             8684          pha             ; save .x
      A7E8  A9 00          8685          lda  #0          ; start temps at zero
      A7EA  85 6F          8686          sta  t0
      A7EC  85 70          8687          sta  t1
      A7EE  85 71          8688          sta  t2
      A7F0  B9 942B        8689          lda  num_sec-1,y
      A7F3  AA             8690          tax             ; number of sectors per track
      A7F4  38             8691  3$      sec
      A7F5  26 6F          8692          rol  t0
      A7F7  26 70          8693          rol  t1
      A7F9  26 71          8694          rol  t2
      A7FB  CA             8695          dex
      A7FC  D0 F6          8696          bne  3$
                           8697
      A7FE  68             8698          pla
      A7FF  AA             8699          tax             ; restore .x
                           8700
      A800  A5 6F          8701          lda  t0          ; write BAM side one
      A802  9D 0146        8702          sta  ovrbuf+$46,x
      A805  A5 70          8703          lda  t1
      A807  9D 0147        8704          sta  ovrbuf+$46+1,x
      A80A  A5 71          8705          lda  t2
      A80C  9D 0148        8706          sta  ovrbuf+$46+2,x
      A80F  E8             8707          inx
      A810  E8             8708          inx
      A811  E8             8709          inx
      A812  E0 33          8710          cpx  #$33        ; skip track 53
      A814  D0 04          8711          bne  4$
                           8712
      A816  E8             8713          inx
      A817  E8             8714          inx
      A818  E8             8715          inx
      A819  C8             8716          iny             ; bypass and leave allocated
                           8717
      A81A  C8             8718  4$      iny
      A81B  C0 24          8719          cpy  #36
      A81D  90 C3          8720          bcc  2$
                           8721
      A81F  20 EEB7        8722          jsr  newmap      ; generate new BAM for side zero
```

```
Error Addr  Code           Seq    Source statement

                           8723
      A822  A0 03         8724            ldy  #3
      A824  A9 80         8725            lda  #$80
      A826  91 6D         8726            sta  (bmpnt),y  ; set double sided bit in BAM side 0
                           8727
      A828  A0 FF         8728            ldy  #$ff
      A82A  A2 22         8729            ldx  #34
      A82C  BD 942C       8730   5$       lda  num_sec,x  ; get # of free sectors
      A82F  91 6D         8731            sta  (bmpnt),y  ; save in top end of BAM zero
      A831  88            8732            dey             ; => $dd-$ff
      A832  CA            8733            dex
      A833  10 F7         8734            bpl  5$
                           8735
      A835  A0 EE         8736            ldy  #$ee       ; offset for track 53
      A837  A9 00         8737            lda  #0
      A839  91 6D         8738            sta  (bmpnt),y  ; allocate track 53
                           8739
      A83B  4C D075       8740            jmp  nfcalc     ; calculate BAM side 0/1
                           8741   ;
                           8742   ;
                           8743   ;-------------------------------------------------
                           8744   ;
                           8745   ;     patch 33
                           8746   ;
      A83E  AD 180F        8747   ptch33  lda  pota1      ; 1/2 Mhz ?
      A841  29 20          8748           and  #$20
      A843  D0 06          8749           bne  1$
                           8750
      A845  20 EFCF        8751   2$      jsr  freuse     ; track < 36 or 1541 mode
      A848  4C EF62        8752           jmp  frets2     ; ret
                           8753
      A84B  A5 80          8754   1$      lda  track      ; is track greater than 35
      A84D  C9 24          8755           cmp  #36
      A84F  90 F4          8756           bcc  2$
                           8757
      A851  20 A4A1        8758           jsr  set_bm     ; setup & update parms
      A854  20 A4E7        8759           jsr  bam_pt     ; find position within table
      A857  D0 19          8760           bne  3$
                           8761
      A859  1D EFE9        8762           ora  bmask,x    ; set it free
      A85C  99 0146        8763           sta  ovrbuf+$46,y
      A85F  20 EF88        8764           jsr  dtybam     ; set dirty flag
      A862  20 A508        8765           jsr  deall_b    ; deallocate BAM
      A865  A5 80          8766           lda  track
      A867  C9 35          8767           cmp  #53        ; skip BAM track side one
      A869  F0 08          8768           beq  4$
                           8769
      A86B  A5 7F          8770           lda  drvnum     ; get drv #
      A86D  0A             8771           asl  a
      A86E  AA             8772           tax             ; *2
```

```
Error Addr  Code           Seq    Source statement

     A86F   4C EF7F       8773            jmp  fre20      ; ret
     A872   38            8774    3$      sec
     A873   60            8775    4$      rts
                          8776    ;
                          8777    ;
                          8778    ;-------------------------------------------------
                          8779    ;
                          8780    ;     patch 34
                          8781    ;
     A874   AD 180F       8782    ptch34  lda  pota1      ; 1/2 Mhz ?
     A877   29 20         8783            and  #$20
     A879   D0 06         8784            bne  1$
                          8785
     A87B   20 EFCF       8786    2$      jsr  freuse     ; track < 36 or 1541 mode
     A87E   4C EF96       8787            jmp  rtch34
                          8788
     A881   A5 80         8789    1$      lda  track      ; check track
     A883   C9 24         8790            cmp  #36
     A885   90 F4         8791            bcc  2$
                          8792
     A887   20 A4A1       8793            jsr  set_bm     ; setup & update parms
     A88A   20 A4E7       8794            jsr  bam_pt     ; find position within the BAM
     A88D   F0 19         8795            beq  3$
                          8796
     A88F   5D EFE9       8797            eor  bmask,x    ; set it used
     A892   99 0146       8798            sta  ovrbuf+$46,y
     A895   20 EF88       8799            jsr  dtybam     ; set dirty flag
     A898   20 A51E       8800            jsr  alloc_b    ; allocate BAM
     A89B   A5 80         8801            lda  track
     A89D   C9 35         8802            cmp  #53        ; skip BAM track side one
     A89F   F0 07         8803            beq  3$
                          8804
     A8A1   A5 7F         8805            lda  drvnum     ; get drv #
     A8A3   0A            8806            asl  a
     A8A4   AA            8807            tax
     A8A5   4C EFB2       8808            jmp  use30
     A8A8   60            8809    3$      rts
                          8810    ;
                          8811    ;
                          8812    ;-------------------------------------------------
                          8813    ;
                          8814    ;     patch 35
                          8815    ;
     A8A9   AD 180F       8816    ptch35  lda  pota1      ; 1/2 Mhz ?
     A8AC   29 20         8817            and  #$20
     A8AE   D0 06         8818            bne  1$
                          8819
     A8B0   20 F011       8820    2$      jsr  setbam     ; track < 36 or 1541 mode
     A8B3   4C F1FD       8821            jmp  rtch35
                          8822
```

```
Error Addr  Code           Seq    Source statement

     A8B6   A5 80         8823   1$      lda  track      ; check track
     A8B8   C9 24         8824          cmp  #36
     A8BA   90 F4         8825          bcc  2$
                          8826
     A8BC   20 A4A1       8827          jsr  set_bm     ; setup & update parms
     A8BF   20 A545       8828          jsr  chk_blk    ; chk alloc w/ bits
                          8829
     A8C2   B9 942C       8830          lda  num_sec,y  ; get number of sectors on this track
     A8C5   8D 024E       8831          sta  lstsec     ; & save it
     A8C8   A5 81         8832   4$      lda  sector     ; what sector are we looking at ?
     A8CA   CD 024E       8833          cmp  lstsec
     A8CD   B0 09         8834          bcs  5$
                          8835
     A8CF   20 A4E7       8836          jsr  bam_pt     ; check availability
     A8D2   D0 06         8837          bne  6$
                          8838
     A8D4   E6 81         8839          inc  sector     ; we will find something I hope...
     A8D6   D0 F0         8840          bne  4$
                          8841
     A8D8   A9 00         8842   5$      lda  #0
     A8DA   60            8843   6$      rts             ; (z=1) used
                          8844   ;
                          8845   ;
                          8846   ;-----------------------------------------------
                          8847   ;
                          8848   ;     patch 36
                          8849   ;
     A8DB   AD 180F       8850   ptch36  lda  pota1     ; 1/2 Mhz ?
     A8DE   29 20         8851          and  #$20
     A8E0   D0 06         8852          bne  1$
                          8853
     A8E2   A5 6F         8854   2$      lda  temp
     A8E4   48            8855          pha
     A8E5   4C F130       8856          jmp  rtch36
                          8857
     A8E8   A5 80         8858   1$      lda  track      ; check track
     A8EA   C9 24         8859          cmp  #36
     A8EC   90 F4         8860          bcc  2$
                          8861
     A8EE   C9 35         8862          cmp  #53
     A8F0   F0 0E         8863          beq  3$
                          8864
     A8F2   A5 6F         8865          lda  temp
     A8F4   48            8866          pha             ; save temp var
     A8F5   20 A4A1       8867          jsr  set_bm     ; setup & update parms
     A8F8   A8            8868          tay             ; save .a
     A8F9   68            8869          pla
     A8FA   85 6F         8870          sta  temp       ; restore temp
     A8FC   98            8871          tya             ; return allocation status in .a
     A8FD   4C F138       8872          jmp  rtch36a
```

```
Error Addr  Code           Seq    Source statement

                           8873
      A900   A9 00         8874   3$       lda  #0          ; z=1
      A902   4C F138       8875            jmp  rtch36a     ; track 53 allocated
                           8876   ;
                           8877   ;
                           8878   ;-------------------------------------------------
                           8879   ;
                           8880   ;     patch 37
                           8881   ;
      A905   AD 180F       8882   ptch37   lda  pota1      ; 1/2 Mhz ?
      A908   29 20         8883            and  #$20
      A90A   D0 06         8884            bne  1$
                           8885
      A90C   20 F011       8886   2$       jsr  setbam      ; setup the BAM pointer
      A90F   4C F1C7       8887            jmp  rtch37      ; ret
                           8888
      A912   A5 80         8889   1$       lda  track       ; check track
      A914   C9 24         8890            cmp  #36
      A916   90 F4         8891            bcc  2$
                           8892
      A918   20 A4A1       8893            jsr  set_bm      ; setup & update parms
      A91B   4C F1C9       8894            jmp  rtch37a
                           8895   ;
                           8896   ;
                           8897   ;-------------------------------------------------
                           8898   ;
                           8899   ;     patch 38
                           8900   ;
      A91E   AD 180F       8901   ptch38   lda  pota1      ; 1/2 Mhz ?
      A921   29 20         8902            and  #$20
      A923   D0 06         8903            bne  1$
                           8904
      A925   20 F011       8905   2$       jsr  setbam      ; setup the BAM pointer
      A928   4C F1E2       8906            jmp  rtch38      ; ret
                           8907
      A92B   A5 80         8908   1$       lda  track       ; check track
      A92D   C9 24         8909            cmp  #36
      A92F   90 F4         8910            bcc  2$
                           8911
      A931   20 A4A1       8912            jsr  set_bm      ; setup & update parms
      A934   4C F1E4       8913            jmp  rtch38a
                           8914   ;
                           8915   ;
                           8916   ;-------------------------------------------------
                           8917   ;
                           8918   ;     patch 39
                           8919   ;
      A937   AD 180F       8920   ptch39   lda  pota1      ; 1/2 Mhz ?
      A93A   29 20         8921            and  #$20
      A93C   D0 03         8922            bne  1$
```

```
Error Addr  Code           Seq    Source statement

                           8923
     A93E   4C F220        8924   2$      jmp  avck        ; check block availability
                           8925
     A941   AD 02AC        8926   1$      lda  maxtrk      ; check maximum track
     A944   C9 25          8927          cmp  #37
     A946   90 F6          8928          bcc  2$
                           8929
     A948   A5 80          8930          lda  track       ; check track do regular check
     A94A   C9 24          8931          cmp  #36
     A94C   90 F0          8932          bcc  2$
                           8933
     A94E   4C A545        8934          jmp  chk_blk     ; finish up
                           8935   ;
                           8936   ;
                           8937   ;-------------------------------------------------
                           8938   ;
                           8939   ;     patch 40
                           8940   ;
     A951   9D 02FA        8941   ptch40  sta  ndbl,x     ; save low order
     A954   AD 180F        8942          lda  pota1       ; 1/2 Mhz ?
     A957   29 20          8943          and  #$20
     A959   F0 23          8944          beq  1$
                           8945
     A95B   AD 02AC        8946          lda  maxtrk      ; check maximum track
     A95E   C9 25          8947          cmp  #37
     A960   90 1C          8948          bcc  1$
                           8949
     A962   20 A48B        8950          jsr  sav_pnt     ; save pointers
     A965   20 A534        8951          jsr  where       ; locate BAM
                           8952
     A968   A0 22          8953          ldy  #34         ; count backwards and sleep........zzzzZZZZZZZZ
     A96A   AD 02FA        8954          lda  ndbl
     A96D   18             8955   2$      clc
     A96E   71 6D          8956          adc  (bmpnt),y   ; count it up
     A970   8D 02FA        8957          sta  ndbl        ; keep track HAHA !!!
     A973   90 03          8958          bcc  3$
                           8959
     A975   EE 02FC        8960          inc  ndbh        ; increment msb
     A978   88             8961   3$      dey
     A979   10 F2          8962          bpl  2$
                           8963
     A97B   4C A496        8964          jmp  res_pnt     ; restore pointers
                           8965
     A97E   60             8966   1$      rts             ; done
                           8967   ;
                           8968   ;
                           8969   ;-------------------------------------------------
                           8970   ;
                           8971   ;     patch 41
                           8972   ;
```

```
Error Addr  Code           Seq    Source statement

     A97F   95 B5         8973   ptch41  sta  nbkl,x
     A981   95 BB         8974           sta  nbkh,x
     A983   A9 00         8975           lda  #0
     A985   9D 0244       8976           sta  lstchr,x
     A988   60            8977           rts
                          8978   ;
                          8979   ;
                          8980   ;-------------------------------------------------
                          8981   ;
                          8982   ;     patch 42
                          8983   ;
     A989   20 A40D       8984   ptch42  jsr  jformat    ; format disk in GCR
     A98C   A0 00         8985           ldy  #0
     A98E   8C 0298       8986           sty  jobrtn
     A991   60            8987           rts
                          8988   ;
            =A99D         8989           *=*+11          ; address adjust
                          8990   ;
                          8991   ;
                          8992   ;-------------------------------------------------
                          8993   ;
                          8994   ;     patch 43
                          8995   ;
                          8996   ;
     A99D   A9 00         8997   ptch43  lda  #0         ; clr nodrv
     A99F                 8998+          NODRWR          ; write nodrv,x absolute
     A99F   9D FF 00      8999A          .byte $9d,$ff,$00
     A9A2   4C C1B7       9000           jmp' rtch43
                          9001   ;
                          9002   ;
                          9003   ;-------------------------------------------------
                          9004   ;
                          9005   ;     patch 44
                          9006   ;
                          9007   ;
     A9A5   98            9008   ptch44  tya             ; set/clr nodrv
     A9A6                 9009+          NODRWR          ; write nodrv,x absolute
     A9A6   9D FF 00      9010A          .byte $9d,$ff,$00
     A9A9   4C C664       9011           jmp  rtch44
                          9012   ;
                          9013   ;
                          9014   ;-------------------------------------------------
                          9015   ;
                          9016   ;     patch 45
                          9017   ;
     A9AC   AD 180F       9018   ptch45  lda  pota1      ; 1/2 Mhz ?
     A9AF   29 20         9019           and  #$20
     A9B1   F0 03         9020           beq  1$
                          9021
     A9B3   4C 815A       9022           jmp  a2          ; 1571 mode
```

```
Error Addr  Code           Seq    Source statement

     A9B6   4C E8D7       9023   1$      jmp  atns20     ; 1541 mode
                          9024   ;
                          9025   ;
                          9026   ;-------------------------------------------------
                          9027   ;
                          9028   ;     patch 46
                          9029   ;
     A9B9   48            9030   ptch46  pha             ; save error
     A9BA   86 F9         9031           stx  jobnum     ; save job #
     A9BC   AD 180F       9032           lda  pota1      ; 1/2 Mhz ?
     A9BF   29 20         9033           and  #$20
     A9C1   F0 0F         9034           beq  1$
                          9035
     A9C3   24 37         9036           bit  fastsr     ; error recovery on ?
     A9C5   10 0B         9037           bpl  1$
                          9038
     A9C7   A5 37         9039           lda  fastsr
     A9C9   29 7F         9040           and  #$7f       ; clr error recovery
     A9CB   85 37         9041           sta  fastsr
                          9042
     A9CD   68            9043           pla             ; get error
     A9CE   AA            9044           tax
     A9CF   4C 9199       9045           jmp  ctr_err    ; let it error out
                          9046
     A9D2   4C E60D       9047   1$      jmp  rtch46     ; return
                          9048   ;
                          9049   ;
                          9050   ;-------------------------------------------------
                          9051   ;
                          9052   ;     patch 47
                          9053   ;
     A9D5   48            9054   ptch47  pha             ; save error
     A9D6   AD 180F       9055           lda  pota1      ; 1/2 Mhz ?
     A9D9   29 20         9056           and  #$20
     A9DB   F0 17         9057           beq  1$
                          9058
     A9DD   24 37         9059           bit  fastsr     ; error recovery on ?
     A9DF   10 13         9060           bpl  1$
                          9061
     A9E1   A5 37         9062           lda  fastsr
     A9E3   29 7F         9063           and  #$7f       ; clr error recovery
     A9E5   85 37         9064           sta  fastsr
     A9E7   78            9065           sei
     A9E8   A2 02         9066           ldx  #2
     A9EA   20 9228       9067           jsr  handsk     ; send it
     A9ED   A9 00         9068           lda  #0
     A9EF   85 83         9069           sta  sa
     A9F1   20 DAC0       9070           jsr  close      ; close channel
                          9071
     A9F4   68            9072   1$      pla             ; get error #
```

```
Error Addr  Code           Seq    Source statement

     A9F5   4C E645        9073            jmp  cmder2
                           9074    ;
                           9075    ;
                           9076    ;-----------------------------------------------------
                           9077    ;
                           9078    ;     patch 48
                           9079    ;
     A9F8   A9 00          9080    ptch48  lda  #0
     A9FA   85 20          9081            sta  drvst        ; clr drive status
     A9FC   AD 1C0C        9082            lda  pcr2         ; set edge and latch mode
     A9FF   4C F266        9083            jmp  rtch48
                           9084    ;
                           9085    ;
                           9086    ;-----------------------------------------------------
                           9087    ;
                           9088    ;     patch 49
                           9089    ;
     AA02   AD 0200        9090    ptch49  lda  cmdbuf       ; is there a 'UOn' cmd waiting in the wings
     AA05   C9 55          9091            cmp  #'U
     AA07   D0 07          9092            bne  1$           ; br, no 'U'
                           9093
     AA09   AD 0201        9094            lda  cmdbuf+1
     AA0C   C9 30          9095            cmp  #'0
     AA0E   F0 04          9096            beq  2$           ; br, we got it toyota
                           9097
     AA10   B9 0200        9098    1$      lda  cmdbuf,y     ; get char
     AA13   2C             9099            .byte skip2
     AA14   A9 00          9100    2$      lda  #0           ; null
     AA16   60             9101            rts
                           9102    ;
                           9103    ;
                           9104    ;-----------------------------------------------------
                           9105    ;
                           9106    ;     patch 50
                           9107    ;
     AA17   A6 7F          9108    ptch50  ldx  drvnum       ; get offset
     AA19                  9109+           NODRRD            ; read nodrv,x absolute
     AA19   BD FF 00       9110A           .byte $bd,$ff,$00
     AA1C   60             9111            rts
                           9112    ;
                           9113    ;
                           9114    ;-----------------------------------------------------
                           9115    ;
                           9116    ;     patch 51
                           9117    ;
     AA1D   95 1C          9118    ptch51  sta  wpsw,x       ; clr wp switch
     AA1F                  9119+           NODRWR            ; write nodrv,x absolute
     AA1F   9D FF 00       9120A           .byte $9d,$ff,$00
     AA22   4C D075        9121            jmp  rtch51
                           9122
```

```
Error Addr  Code           Seq    Source statement

                          9123   ;
                          9124   ;-------------------------------------------------
                          9125   ;
                          9126   ;     patch 52
                          9127   ;
     AA25   A6 7F         9128   ptch52  ldx  drvnum      ; get offset
     AA27                 9129+          NODRRD           ; read nodrv,x absolute
     AA27   BD FF 00      9130A          .byte $bd,$ff,$00
     AA2A   4C F01B       9131           jmp  rtch52
                          9132   ;
                          9133   ;
                          9134   ;-------------------------------------------------
                          9135   ;
                          9136   ;     patch 53
                          9137   ;
     AA2D   A5 75         9138   ptch53  lda  ip
     AA2F   C9 67         9139           cmp  #<sysirq    ; irq ?
     AA31   D0 09         9140           bne  1$
                          9141
     AA33   A5 76         9142           lda  ip+1
     AA35   C9 FE         9143           cmp  #>sysirq    ; irq ?
     AA37   D0 03         9144           bne  1$
                          9145
     AA39   00            9146           brk              ; do irq and return
     AA3A   EA            9147           nop
     AA3B   60            9148           rts
                          9149
     AA3C   6C 0075       9150   1$      jmp  (ip)
                          9151           .include  newvec
```

```
Error Addr  Code           Seq    Source statement

                           9153           .subttl 'newvec.src'
                           9154
            =BF00          9155           *=$BF00
                           9156           ; vector table
                           9157
      BF00  4C 9D88        9158           jmp  irq
      BF03  4C 9DDE        9159           jmp  jirq
      BF06  4C F2B0        9160           jmp  icc
      BF09  4C 92BA        9161           jmp  jicc
      BF0C  4C F393        9162           jmp  setjb
      BF0F  4C 93D1        9163           jmp  jsetjb
      BF12  4C F969        9164           jmp  errr
      BF15  4C 99B5        9165           jmp  jerrr
      BF18  4C FE00        9166           jmp  kill
      BF1B  4C F934        9167           jmp  conhdr
      BF1E  4C F556        9168           jmp  sync
      BF21  4C 9754        9169           jmp  jsync
      BF24  4C F8E0        9170           jmp  gcrbin
      BF27  4C 9965        9171           jmp  jgcrbin
      BF2A  4C F5E9        9172           jmp  chkblk
      BF2D  4C F7E6        9173           jmp  get4gb
      BF30  4C 98D9        9174           jmp  jget4gb
      BF33  4C A483        9175           jmp  jslowd
      BF36  4C FEF3        9176           jmp  slowd
      BF39  4C A47E        9177           jmp  jslower
      BF3C  4C F005        9178           jmp  clrbam
      BF3F  4C F0D1        9179           jmp  clnbam
      BF42  4C C146        9180           jmp  parsxq
      BF45  4C C268        9181           jmp  parse
      BF48  4C C2B3        9182           jmp  cmdset
      BF4B  4C C2DC        9183           jmp  cmdrst
      BF4E  4C 86E6        9184           jmp  prcmd
      BF51  4C 8764        9185           jmp  moton
      BF54  4C 8770        9186           jmp  motoff
      BF57  4C 808E        9187           jmp  cmdtbb
      BF5A  4C CF1E        9188           jmp  dblbuf
      BF5D  4C D7B4        9189           jmp  open
      BF60  4C DAC0        9190           jmp  close
      BF63  4C E60A        9191           jmp  error
      BF66  4C 9080        9192           jmp  fstload
      BF69  4C 924E        9193           jmp  signature
      BF6C  4C F259        9194           jmp  cntint
      BF6F  4C F99C        9195           jmp  end
      BF72  4C 99CA        9196           jmp  jend
      BF75  4C FE95        9197           jmp  cjumpl
                           9198
            =BF96          9199           *=*+30          ; more room for vector jumps
                           9200           ; info tables
                           9201
      BF96  86C8           9202           .word cmdjmp
```

```
Error Addr  Code           Seq    Source statement

                          9203
```

```
Error Addr  Code          Seq   Source statement

                         9205           .subttl '1541 DOS & controller routines'
```

```
Error Addr  Code          Seq    Source statement

                         9207           .include  romsf
```

```
Error Addr  Code           Seq    Source statement

                          9209            .subttl 'romsf.src'
            =C000         9210    *=rom                   ; +$300 rom patch area
     C000   FF             9211    cchksm   .byte  $ff     ; <<<<< TO BE DETERMINED
     C001   =C100          9212    freec0   *=*+255        ; c0 patch space
                          9213            .include  leds
```

```
Error Addr  Code           Seq    Source statement

                           9215           .subttl 'leds.src'
                           9216
                           9217   ;turn on activity led specified
                           9218   ; by drvnum
                           9219
     C100   78             9220   setlds  sei
     C101   A9 F7          9221           lda  #$ff-led1-led0
     C103   2D 1C00        9222           and  ledprt
     C106   48             9223           pha
                           9224
     C107   A5 7F          9225           lda  drvnum
     C109   F0 05          9226           beq  leds0
     C10B   68             9227           pla
     C10C   09 00          9228           ora  #led1
     C10E   D0 03          9229           bne  leds1
     C110                  9230   leds0
     C110   68             9231           pla
     C111   09 08          9232           ora  #led0
     C113                  9233   leds1
     C113   8D 1C00        9234           sta  ledprt
     C116   58             9235           cli
     C117   60             9236           rts
                           9237
     C118   78             9238   ledson  sei
     C119   A9 08          9239           lda  #led1+led0
     C11B   0D 1C00        9240           ora  ledprt
     C11E   8D 1C00        9241           sta  ledprt
     C121   58             9242           cli
     C122   60             9243           rts
                           9244
     C123                  9245   erroff
     C123   A9 00          9246           lda  #0
     C125   8D 026C        9247           sta  erword
     C128   8D 026D        9248           sta  erled
     C12B   60             9249           rts
                           9250
     C12C   78             9251   erron   sei
     C12D   8A             9252           txa
     C12E   48             9253           pha                ; save .x
     C12F   A9 50          9254           lda  #80
     C131   8D 026C        9255           sta  erword
     C134   A2 00          9256           ldx  #0
     C136   BD FECA        9257           lda  ledmsk,x
     C139   8D 026D        9258           sta  erled
     C13C   0D 1C00        9259           ora  ledprt        ; set led on
     C13F   8D 1C00        9260           sta  ledprt
     C142   68             9261           pla
     C143   AA             9262           tax                ; restore .x
     C144   58             9263           cli
     C145   60             9264           rts
```

```
Error Addr  Code           Seq    Source statement

                          9265            .include  parsex
```

```
Error Addr  Code           Seq    Source statement

                           9267            .subttl 'parsex.src'
                           9268
                           9269    ;parse & execute string in cmdbuf
                           9270
     C146                  9271    parsxq
     C146   A9 00          9272            lda  #0
     C148   8D 02F9        9273            sta  wbam
     C14B   AD 028E        9274            lda  lstdrv
     C14E   85 7F          9275            sta  drvnum
     C150   20 E6BC        9276            jsr  okerr
     C153   A5 84          9277            lda  orgsa
     C155   10 09          9278            bpl  ps05
     C157   29 0F          9279            and  #$f
     C159   C9 0F          9280            cmp  #$f
     C15B   F0 03          9281            beq  ps05
     C15D   4C D7B4        9282            jmp  open
     C160   20 C2B3        9283    ps05    jsr  cmdset     ; set variables,regs
     C163   B1 A3          9284            lda  (cb),y
     C165   8D 0275        9285            sta  char
     C168   A2 0B          9286            ldx  #ncmds-1   ; search cmd table
     C16A   BD FE89        9287    ps10    lda  cmdtbl,x
     C16D   CD 0275        9288            cmp  char
     C170   F0 08          9289            beq  ps20
     C172   CA             9290            dex
     C173   10 F5          9291            bpl  ps10
     C175   A9 31          9292            lda  #badcmd    ; no such cmd
     C177   4C C1C8        9293            jmp  cmderr
     C17A   8E 022A        9294    ps20    stx  cmdnum     ; x= cmd #
                           9295
     C17D   E0 09          9296            cpx  #pcmd      ; cmds not parsed
     C17F   90 03          9297            bcc  ps30
     C181   20 C1EE        9298            jsr  tagcmd     ; set tables, pointers &patterns
     C184   AE 022A        9299    ps30    ldx  cmdnum
     C187   BD FE95        9300            lda  cjumpl,x
     C18A   85 6F          9301            sta  temp
     C18C   BD FEA1        9302            lda  cjumph,x
     C18F   85 70          9303            sta  temp+1
     C191   6C 006F        9304            jmp  (temp)     ; command table jump
                           9305
                           9306    ; successful command termination
     C194                  9307    endcmd
     C194   A9 00          9308            lda  #0
     C196   8D 02F9        9309            sta  wbam
     C199                  9310    endsav
     C199   AD 026C        9311            lda  erword
     C19C   D0 2A          9312            bne  cmderr
                           9313    ;
     C19E   A0 00          9314            ldy  #0
     C1A0   98             9315            tya
     C1A1   84 80          9316            sty  track
```

```
Error Addr  Code           Seq    Source statement

                           9317
      C1A3   84 81         9318   scrend  sty  sector     ; scratch entry
      C1A5   84 A3         9319           sty  cb
      C1A7   20 E6C7       9320           jsr  errmsg
      C1AA   20 C123       9321           jsr  erroff
      C1AD                 9322   scren1
      C1AD   A5 7F         9323           lda  drvnum
      C1AF   8D 028E       9324           sta  lstdrv
      C1B2   AA            9325           tax
                           9326
                           9327   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     C1DB   F0 05         9367            beq  sp10
     C1DD   88            9368            dey
     C1DE   88            9369            dey
     C1DF   8C 027A       9370            sty  filtbl
     C1E2   4C C368       9371   sp10     jmp  setany      ; set drive #
```

```
Error Addr  Code           Seq   Source statement

     C1E5                 9373   prscln
     C1E5   A0 00         9374          ldy  #0
     C1E7   A2 00         9375          ldx  #0
     C1E9   A9 3A         9376          lda  #':
     C1EB   4C C268       9377          jmp  parse      ; find pos'n of ":"
                          9378   ;
                          9379   ;tag command string
                          9380   ;  set up cmd structure
                          9381   ;  image & file stream ptrs
                          9382   ;
                          9383
     C1EE                 9384   tagcmd
     C1EE   20 C1E5       9385          jsr  prscln
     C1F1   D0 05         9386          bne  tc30
     C1F3   A9 34         9387   tc25   lda  #nofile    ; none, no files
     C1F5   4C C1C8       9388          jmp  cmderr
     C1F8   88            9389   tc30   dey
     C1F9   88            9390          dey
     C1FA   8C 027A       9391          sty  filtbl     ; ":"-1 starts fs1
     C1FD   8A            9392          txa
     C1FE   D0 F3         9393          bne  tc25       ; err: "," before ":"
     C200   A9 3D         9394   tc35   lda  #'=        ; search: "="
     C202   20 C268       9395          jsr  parse
     C205   8A            9396          txa             ; ?file count= 1-1?
     C206   F0 02         9397          beq  tc40
     C208   A9 40         9398          lda  #%01000000 ; g1-bit
     C20A   09 21         9399   tc40   ora  #%00100001 ; e1,^e2-bits
     C20C   8D 028B       9400          sta  image      ; fs structure
     C20F   E8            9401          inx
     C210   8E 0277       9402          stx  f1cnt
     C213   8E 0278       9403          stx  f2cnt      ; init for no fs2
     C216   AD 028A       9404          lda  patflg
     C219   F0 0D         9405          beq  tc50
     C21B   A9 80         9406          lda  #%10000000 ; p1-bit
     C21D   0D 028B       9407          ora  image
     C220   8D 028B       9408          sta  image
     C223   A9 00         9409          lda  #0
     C225   8D 028A       9410          sta  patflg     ; clear pattern flag
     C228   98            9411   tc50   tya             ; ptr to fs2
     C229   F0 29         9412          beq  tc75       ;  fs2 not here
     C22B   9D 027A       9413          sta  filtbl,x
     C22E   AD 0277       9414          lda  f1cnt      ; fs2 is here now,...
     C231   8D 0279       9415          sta  f2ptr      ; ...now set f2 ptr
     C234   A9 8D         9416          lda  #$8d       ; find cr-shifted
     C236   20 C268       9417          jsr  parse      ; parse rest of cmd string
     C239   E8            9418          inx             ; advance filtbl ptr to end
     C23A   8E 0278       9419          stx  f2cnt      ; save it
     C23D   CA            9420          dex             ; restore for test
     C23E   AD 028A       9421          lda  patflg     ; save last pattern
     C241   F0 02         9422          beq  tc60       ; ?any patterns?
```

```
Error Addr  Code           Seq    Source statement

     C243   A9 08         9423            lda  #%1000      ; yes, p2-bit
     C245   EC 0277       9424    tc60    cpx  f1cnt       ; ?f2cnt=f1cnt+1?
     C248   F0 02         9425            beq  tc70
     C24A   09 04         9426            ora  #%0100      ; g2-bit
     C24C   09 03         9427    tc70    ora  #%0011      ; e2-bit,^e2-bit
     C24E   4D 028B       9428            eor  image       ; eor clears ^e2-bit
     C251   8D 028B       9429            sta  image
     C254                 9430    tc75
     C254   AD 028B       9431            lda  image
     C257   AE 022A       9432            ldx  cmdnum
     C25A   3D FEA5       9433            and  struct,x    ; match cmd template
     C25D   D0 01         9434            bne  tc80
     C25F   60            9435            rts
     C260   8D 026C       9436    tc80    sta  erword      ; **could be warning
     C263   A9 30         9437            lda  #badsyn     ; err: bad syntax
     C265   4C C1C8       9438            jmp  cmderr
                          9439
```

```
Error Addr  Code           Seq    Source statement

                           9441            .subttl  'parse'
                           9442
                           9443    ;parse string
                           9444    ;  looks for special chars,
                           9445    ;  returning when var'bl char
                           9446    ;  is found
                           9447    ;   a: var'bl char
                           9448    ;   x: in,out: index, filtbl+1
                           9449    ;   y: in: index, cmdbuf
                           9450    ;     out: new ptr, =0 if none
                           9451    ;          (z=1) if y=0
                           9452
     C268   8D 0275        9453    parse   sta  char        ; save var'bl char
     C26B   CC 0274        9454    pr10    cpy  cmdsiz      ; stay in string
     C26E   B0 2E          9455            bcs  pr30
     C270   B1 A3          9456            lda  (cb),y      ; match char
     C272   C8             9457            iny
     C273   CD 0275        9458            cmp  char
     C276   F0 28          9459            beq  pr35        ; found char
     C278   C9 2A          9460            cmp  #'*         ; match pattern chars
     C27A   F0 04          9461            beq  pr20
     C27C   C9 3F          9462            cmp  #'?
     C27E   D0 03          9463            bne  pr25
     C280   EE 028A        9464    pr20    inc  patflg      ; set pattern flag
     C283   C9 2C          9465    pr25    cmp  #',         ; match file separator
     C285   D0 E4          9466            bne  pr10
     C287   98             9467            tya
     C288   9D 027B        9468            sta  filtbl+1,x  ; put ptrs in table
     C28B   AD 028A        9469            lda  patflg      ; save pattern for ea file
     C28E   29 7F          9470            and  #$7f
     C290   F0 07          9471            beq  pr28
     C292   A9 80          9472            lda  #$80        ; retain pattern presence...
     C294   95 E7          9473            sta  pattyp,x
     C296   8D 028A        9474            sta  patflg      ; ...but clear count
     C299   E8             9475    pr28    inx
     C29A   E0 04          9476            cpx  #mxfils-1
     C29C   90 CD          9477            bcc  pr10        ; no more than mxfils
     C29E   A0 00          9478    pr30    ldy  #0          ; y=0 (z=1)
     C2A0   AD 0274        9479    pr35    lda  cmdsiz
     C2A3   9D 027B        9480            sta  filtbl+1,x
     C2A6   AD 028A        9481            lda  patflg
     C2A9   29 7F          9482            and  #$7f
     C2AB   F0 04          9483            beq  pr40
     C2AD   A9 80          9484            lda  #$80
     C2AF   95 E7          9485            sta  pattyp,x
     C2B1   98             9486    pr40    tya              ; z is set
     C2B2   60             9487            rts
                           9488
                           9489    ;initialize command tables, ptrs, etc.
                           9490
```

```
Error Addr  Code           Seq   Source statement

     C2B3   A4 A3         9491   cmdset  ldy  buftab+cbptr
     C2B5   F0 14         9492           beq  cs08
     C2B7   88            9493           dey
     C2B8   F0 10         9494           beq  cs07
                          9495
                          9496   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     C2FD   9D 0279       9541   cs10    sta  filtbl-1,x
     C300   95 D7         9542           sta  entsec-1,x
     C302   95 DC         9543           sta  entind-1,x
     C304   95 E1         9544           sta  fildrv-1,x
     C306   95 E6         9545           sta  pattyp-1,x
     C308   9D 027F       9546           sta  filtrk-1,x
     C30B   9D 0284       9547           sta  filsec-1,x
     C30E   CA            9548           dex
     C30F   D0 EC         9549           bne  cs10
     C311   60            9550           rts
                          9551           .include  setdrv
```

```
Error Addr  Code           Seq   Source statement

                           9553          .subttl  'setdrv.src'
                           9554  ;set 1st drive and table pointers
     C312   AD 0278        9555  onedrv  lda  f2cnt
     C315   8D 0277        9556          sta  f1cnt
     C318   A9 01          9557          lda  #1
     C31A   8D 0278        9558          sta  f2cnt
     C31D   8D 0279        9559          sta  f2ptr
                           9560
                           9561  ;set up all drives from f2cnt
                           9562
     C320   AC 028E        9563  alldrs  ldy  lstdrv     ; set up drive #'s...
     C323   A2 00          9564          ldx  #0         ; ...into file entry table...
     C325   86 D3          9565  ad10    stx  f1ptr      ; ...on sector ptr byte
     C327   BD 027A        9566          lda  filtbl,x
     C32A   20 C33C        9567          jsr  setdrv
     C32D   A6 D3          9568          ldx  f1ptr          .
     C32F   9D 027A        9569          sta  filtbl,x   ; incr ptr past ":"
     C332   98             9570          tya             ; bits rep drives
     C333   95 E2          9571          sta  fildrv,x   ; bit7: default
     C335   E8             9572          inx             ; bit0: drive #
     C336   EC 0278        9573          cpx  f2cnt
     C339   90 EA          9574          bcc  ad10
     C33B   60             9575          rts
                           9576
                           9577  ;  set drive number
                           9578  ;  determines drive # from text or
                           9579  ;  uses default (-d)
                           9580  ;  a: in,out: index, cmdbuf
                           9581  ;  y: in: default drive
                           9582  ;  out: drive number, - if default
                           9583
     C33C   AA              9584  setdrv  tax             ; x= cmdbuf index
     C33D   A0 00           9585          ldy  #0         ; set default drive to zero!!!!!!!!!!!!!!!!!!!!!
     C33F   A9 3A           9586          lda  #':
     C341   DD 0201         9587          cmp  cmdbuf+1,x ; for xxx:file
     C344   F0 0C           9588          beq  sd40       ;      ^
     C346   DD 0200         9589          cmp  cmdbuf,x   ; for xxx:file
                           9590
     C349   D0 16           9591          bne  sd50       ;        ^
                           9592
     C34B   E8             9593          inx             ; found ":", so...
     C34C   98             9594  sd20    tya             ; drive= default
     C34D   29 01          9595  sd22    and  #1         ; convert to numeric
     C34F   A8             9596  sd24    tay             ; restore drive
     C350   8A             9597          txa             ; a=index & xxxxfile
     C351   60             9598          rts             ;                 ^
                           9599
     C352   BD 0200        9600  sd40    lda  cmdbuf,x
     C355   E8             9601          inx             ;  xxx:file
     C356   E8             9602          inx             ;    --^
```

```
Error Addr  Code          Seq   Source statement

     C357   C9 30        9603          cmp  #'0        ; for xx0:file
     C359   F0 F2        9604          beq  sd22       ;          ^
                         9605
     C35B   C9 31        9606          cmp  #'1        ; for xx1:file
     C35D   F0 EE        9607          beq  sd22       ;          ^
                         9608
     C35F   D0 EB        9609          bne  sd20       ; cmd:file or xx,:file
                         9610
            =C361        9611  sd50     =*             ;     ^           ^
     C361   98           9612          tya             ; for xxx,file or xx=file
     C362   09 80        9613          ora  #$80       ;         ^           ^
     C364   29 81        9614          and  #$81       ; drive= -default
     C366   D0 E7        9615          bne  sd24       ; finish testing
                         9616
                         9617  ;set drive from any config.
                         9618
     C368   A9 00        9619  setany  lda  #0
     C36A   8D 028B      9620          sta  image
     C36D   AC 027A      9621          ldy  filtbl
     C370   B1 A3        9622  sa05    lda  (cb),y
     C372   20 C3BD      9623          jsr  tst0v1
     C375   10 11        9624          bpl  sa20
     C377   C8           9625          iny
     C378   CC 0274      9626          cpy  cmdsiz
     C37B   B0 06        9627          bcs  sa10
     C37D   AC 0274      9628          ldy  cmdsiz
     C380   88           9629          dey
     C381   D0 ED        9630          bne  sa05
     C383   CE 028B      9631  sa10    dec  image
     C386   A9 00        9632          lda  #0
                         9633
                         9634  ;default to zero
                         9635
     C388   29 01        9636  sa20    and  #1
     C38A   85 7F        9637          sta  drvnum
     C38C   4C C100      9638          jmp  setlds
                         9639
                         9640  ;toggle drvnum
                         9641
     C38F   A5 7F        9642  togdrv  lda  drvnum
     C391   49 01        9643          eor  #1
     C393   29 01        9644          and  #1
     C395   85 7F        9645          sta  drvnum
     C397   60           9646          rts
                         9647
                         9648  ;set ptrs to one file stream & chk type
                         9649
     C398   A0 00        9650  fs1set  ldy  #0
     C39A   AD 0277      9651          lda  f1cnt
     C39D   CD 0278      9652          cmp  f2cnt
```

```
Error Addr  Code           Seq    Source statement

     C3A0   F0 16         9653            beq  fs15
     C3A2   CE 0278       9654            dec  f2cnt
     C3A5   AC 0278       9655            ldy  f2cnt
     C3A8   B9 027A       9656            lda  filtbl,y
     C3AB   A8            9657            tay
     C3AC   B1 A3         9658            lda  (cb),y
     C3AE   A0 04         9659            ldy  #ntypes-1
     C3B0   D9 FEBB       9660   fs10     cmp  typlst,y
     C3B3   F0 03         9661            beq  fs15
     C3B5   88            9662            dey
     C3B6   D0 F8         9663            bne  fs10
     C3B8   98            9664   fs15     tya
     C3B9   8D 0296       9665            sta  typflg
     C3BC   60            9666            rts
                          9667
                          9668   ;test char in accum for "0" or "1"
                          9669
     C3BD   C9 30         9670   tst0v1   cmp  #'0
     C3BF   F0 06         9671            beq  t0v1
     C3C1   C9 31         9672            cmp  #'1
     C3C3   F0 02         9673            beq  t0v1
     C3C5   09 80         9674            ora  #$80
     C3C7   29 81         9675   t0v1     and  #$81
     C3C9   60            9676            rts
                          9677            .include  lookup
```

```
Error Addr  Code           Seq   Source statement

                           9679           .subttl 'lookup.src'
                           9680   ;  optsch  optimal search for lookup
                           9681   ;  and fndfil
                           9682
     C3CA   A9 00          9683   optsch  lda  #0          ; determine optimal search
     C3CC   85 6F          9684           sta  temp        ; init drive mask
     C3CE   8D 028D        9685           sta  drvflg
     C3D1   48             9686           pha
     C3D2   AE 0278        9687           ldx  f2cnt
     C3D5   68             9688   os10    pla
     C3D6   05 6F          9689           ora  temp
     C3D8   48             9690           pha
     C3D9   A9 01          9691           lda  #1
     C3DB   85 6F          9692           sta  temp
     C3DD   CA             9693           dex
     C3DE   30 0F          9694           bmi  os30
     C3E0   B5 E2          9695           lda  fildrv,x
     C3E2   10 04          9696           bpl  os15
     C3E4   06 6F          9697           asl  temp
     C3E6   06 6F          9698           asl  temp
                           9699
     C3E8   4A             9700   os15    lsr  a
     C3E9   90 EA          9701           bcc  os10
     C3EB   06 6F          9702           asl  temp
     C3ED   D0 E6          9703           bne  os10        ; (branch)
                           9704
     C3EF   68             9705   os30    pla
     C3F0   AA             9706           tax
     C3F1   BD C43F        9707           lda  schtbl-1,x
     C3F4   48             9708           pha
     C3F5   29 03          9709           and  #3
     C3F7   8D 028C        9710           sta  drvcnt
     C3FA   68             9711           pla
     C3FB   0A             9712           asl  a
     C3FC   10 3E          9713           bpl  os40
     C3FE   A5 E2          9714           lda  fildrv
     C400   29 01          9715   os35    and  #1
     C402   85 7F          9716           sta  drvnum
                           9717
     C404   AD 028C        9718           lda  drvcnt
     C407   F0 2B          9719           beq  os60        ; only one drive addressed
                           9720
     C409   20 C63D        9721           jsr  autoi       ; check drive for autoinit
     C40C   F0 12          9722           beq  os50        ; drive is active
                           9723
     C40E   20 C38F        9724           jsr  togdrv
     C411   A9 00          9725           lda  #0          ; set 1 drive addressed
     C413   8D 028C        9726           sta  drvcnt
     C416   20 C63D        9727           jsr  autoi       ; check drive for autoinit
     C419   F0 1E          9728           beq  os70        ; drive is active
```

```
Error Addr  Code           Seq   Source statement

     C41B                9729  os45
     C41B   A9 74        9730          lda  #nodriv
     C41D   20 C1C8      9731          jsr  cmderr
     C420                9732  os50
     C420   20 C38F      9733          jsr  togdrv
     C423   20 C63D      9734          jsr  autoi      ; check drive for autoinit
     C426   08           9735          php
     C427   20 C38F      9736          jsr  togdrv
     C42A   28           9737          plp
     C42B   F0 0C        9738          beq  os70       ; drive is active
                         9739
     C42D   A9 00        9740          lda  #0         ; set 1 drive addressed
     C42F   8D 028C      9741          sta  drvcnt
     C432   F0 05        9742          beq  os70       ; bra
     C434                9743  os60
     C434   20 C63D      9744          jsr  autoi      ; check drive for autoinit
     C437   D0 E2        9745          bne  os45       ; drive is not active
     C439                9746  os70
     C439   4C C100      9747          jmp  setlds
                         9748
     C43C   2A           9749  os40    rol  a
     C43D   4C C400      9750          jmp  os35
                         9751
     C440   00 80 41     9752  schtbl  .byte    0,$80,$41
     C443   01 01 01     9753          .byte    1,1,1,1
     C446   01
     C447   81 81 81     9754          .byte    $81,$81,$81,$81
     C44A   81
     C44B   42 42 42     9755          .byte    $42,$42,$42,$42
     C44E   42
```

```
Error Addr  Code           Seq   Source statement

                           9757  ;   look up all files in stream
                           9758  ;   and fill tables w/ info
                           9759
     C44F   20 C3CA        9760  lookup  jsr  optsch
     C452   A9 00          9761  lk05    lda  #0
     C454   8D 0292        9762          sta  delind
     C457   20 C5AC        9763          jsr  srchst      ; start search
     C45A   D0 19          9764          bne  lk25
     C45C   CE 028C        9765  lk10    dec  drvcnt
     C45F   10 01          9766          bpl  lk15
     C461   60             9767          rts              ; no more drive searches
                           9768
     C462   A9 01          9769  lk15    lda  #1          ; toggle drive #
     C464   8D 028D        9770          sta  drvflg
     C467   20 C38F        9771          jsr  togdrv
     C46A   20 C100        9772          jsr  setlds      ;  turn on led
     C46D   4C C452        9773          jmp  lk05
     C470   20 C617        9774  lk20    jsr  search      ; find valid fn
     C473   F0 10          9775          beq  lk30        ; end of search
     C475   20 C4D8        9776  lk25    jsr  compar      ; compare dir w/ table
     C478   AD 028F        9777          lda  found       ; found flag
     C47B   F0 01          9778          beq  lk26        ; all fn's not found, yet
     C47D   60             9779          rts
                           9780
     C47E   AD 0253        9781  lk26    lda  entfnd
     C481   30 ED          9782          bmi  lk20
     C483   10 F0          9783          bpl  lk25
                           9784
     C485   AD 028F        9785  lk30    lda  found
     C488   F0 D2          9786          beq  lk10
     C48A   60             9787          rts
                           9788
                           9789  ;  find next file name matching
                           9790  ;  any file in stream & return
                           9791  ;  with entry found stuffed into
                           9792  ;  tables
                           9793
     C48B   20 C604        9794  ffre    jsr  srre        ; find file re-entry
     C48E   F0 1A          9795          beq  ff10
     C490   D0 28          9796          bne  ff25
                           9797
     C492   A9 01          9798  ff15    lda  #1
     C494   8D 028D        9799          sta  drvflg
     C497   20 C38F        9800          jsr  togdrv
     C49A   20 C100        9801          jsr  setlds
                           9802
     C49D   A9 00          9803  ffst    lda  #0          ; find file start entry
     C49F   8D 0292        9804          sta  delind
     C4A2   20 C5AC        9805          jsr  srchst
     C4A5   D0 13          9806          bne  ff25
```

```
Error Addr  Code           Seq    Source statement

     C4A7   8D 028F       9807            sta  found
     C4AA   AD 028F       9808    ff10    lda  found
     C4AD   D0 28         9809            bne  ff40
     C4AF   CE 028C       9810            dec  drvcnt
     C4B2   10 DE         9811            bpl  ff15
     C4B4   60            9812            rts
                          9813
     C4B5   20 C617       9814    fndfil  jsr  search      ; find file continuous...
     C4B8   F0 F0         9815            beq  ff10        ; ... re-entry, no channel activity
     C4BA   20 C4D8       9816    ff25    jsr  compar      ; compare file names
     C4BD   AE 0253       9817            ldx  entfnd
     C4C0   10 07         9818            bpl  ff30
     C4C2   AD 028F       9819            lda  found
     C4C5   F0 EE         9820            beq  fndfil
     C4C7   D0 0E         9821            bne  ff40
                          9822
     C4C9   AD 0296       9823    ff30    lda  typflg
     C4CC   F0 09         9824            beq  ff40        ; no type restriction
     C4CE   B5 E7         9825            lda  pattyp,x
     C4D0   29 07         9826            and  #typmsk
     C4D2   CD 0296       9827            cmp  typflg
     C4D5   D0 DE         9828            bne  fndfil
     C4D7   60            9829    ff40    rts
```

```
Error Addr  Code           Seq    Source statement

                           9831
                           9832   ;  compare all filenames in stream table
                           9833   ;  with each valid entry in the
                           9834   ;  directory.  matches are tabulated
                           9835
     C4D8   A2 FF          9836   compar  ldx  #$ff
     C4DA   8E 0253        9837           stx  entfnd
     C4DD   E8             9838           inx
     C4DE   8E 028A        9839           stx  patflg
     C4E1   20 C589        9840           jsr  cmpchk
     C4E4   F0 06          9841           beq  cp10
     C4E6   60             9842   cp02    rts             ; all are found
                           9843
     C4E7   20 C594        9844   cp05    jsr  cc10
     C4EA   D0 FA          9845           bne  cp02
     C4EC   A5 7F          9846   cp10    lda  drvnum
     C4EE   55 E2          9847           eor  fildrv,x
     C4F0   4A             9848           lsr  a
     C4F1   90 0B          9849           bcc  cp20       ; right drive
     C4F3   29 40          9850           and  #$40
     C4F5   F0 F0          9851           beq  cp05       ; no default
     C4F7   A9 02          9852           lda  #2
     C4F9   CD 028C        9853           cmp  drvcnt
     C4FC   F0 E9          9854           beq  cp05       ; don't use default
                           9855
     C4FE   BD 027A        9856   cp20    lda  filtbl,x   ; good drive match
     C501   AA             9857           tax
     C502   20 C6A6        9858           jsr  fndlmt
     C505   A0 03          9859           ldy  #3
     C507   4C C51D        9860           jmp  cp33
     C50A                  9861   cp30
     C50A   BD 0200        9862           lda  cmdbuf,x
     C50D   D1 94          9863           cmp  (dirbuf),y
     C50F   F0 0A          9864           beq  cp32       ; chars are =
                           9865
     C511   C9 3F          9866           cmp  #'?
     C513   D0 D2          9867           bne  cp05       ; no single pattern
     C515   B1 94          9868           lda  (dirbuf),y
     C517   C9 A0          9869           cmp  #$a0
     C519   F0 CC          9870           beq  cp05       ; end of filename
     C51B                  9871   cp32
     C51B   E8             9872           inx
     C51C   C8             9873           iny
     C51D                  9874   cp33
     C51D   EC 0276        9875           cpx  limit
     C520   B0 09          9876           bcs  cp34       ; end of pattern
                           9877
     C522   BD 0200        9878           lda  cmdbuf,x
     C525   C9 2A          9879           cmp  #'*
     C527   F0 0C          9880           beq  cp40       ; star matches all
```

```
Error Addr  Code           Seq    Source statement

     C529   DO DF         9881           bne  cp30        ; keep checking
     C52B                 9882   cp34
     C52B   CO 13         9883           cpy  #19
     C52D   BO 06         9884           bcs  cp40        ; end of filename
                          9885
     C52F   B1 94         9886           lda  (dirbuf),y
     C531   C9 AO         9887           cmp  #$a0
     C533   DO B2         9888           bne  cp05
                          9889
     C535   AE 0279       9890   cp40    ldx  f2ptr       ; filenames match
     C538   8E 0253       9891           stx  entfnd
     C53B   B5 E7         9892           lda  pattyp,x    ; store info in tables
     C53D   29 80         9893           and  #$80
     C53F   8D 028A       9894           sta  patflg
     C542   AD 0294       9895           lda  index
     C545   95 DD         9896           sta  entind,x
     C547   A5 81         9897           lda  sector
     C549   95 D8         9898           sta  entsec,x
     C54B   AO 00         9899           ldy  #0
     C54D   B1 94         9900           lda  (dirbuf),y
     C54F   C8            9901           iny
     C550   48            9902           pha
     C551   29 40         9903           and  #$40
     C553   85 6F         9904           sta  temp
     C555   68            9905           pla
     C556   29 DF         9906           and  #$ff-$20
     C558   30 02         9907           bmi  cp42
                          9908
     C55A   09 20         9909           ora  #$20
     C55C                 9910   cp42
     C55C   29 27         9911           and  #$27
     C55E   05 6F         9912           ora  temp
     C560   85 6F         9913           sta  temp
     C562   A9 80         9914           lda  #$80
     C564   35 E7         9915           and  pattyp,x
     C566   05 6F         9916           ora  temp
     C568   95 E7         9917           sta  pattyp,x
     C56A   B5 E2         9918           lda  fildrv,x
     C56C   29 80         9919           and  #$80
     C56E   05 7F         9920           ora  drvnum
     C570   95 E2         9921           sta  fildrv,x
                          9922
     C572   B1 94         9923           lda  (dirbuf),y
     C574   9D 0280       9924           sta  filtrk,x
     C577   C8            9925           iny
     C578   B1 94         9926           lda  (dirbuf),y
     C57A   9D 0285       9927           sta  filsec,x
     C57D   AD 0258       9928           lda  rec
     C580   DO 07         9929           bne  cp50
     C582   AO 15         9930           ldy  #21
```

```
Error Addr  Code           Seq    Source statement

     C584   B1 94         9931            lda  (dirbuf),y
     C586   8D 0258       9932            sta  rec
                          9933
                          9934    ;check table for unfound files
                          9935
     C589                 9936    cp50
     C589   A9 FF         9937    cmpchk  lda  #$ff
     C58B   8D 028F       9938            sta  found
     C58E   AD 0278       9939            lda  f2cnt
     C591   8D 0279       9940            sta  f2ptr
                          9941
     C594   CE 0279       9942    cc10    dec  f2ptr
     C597   10 01         9943            bpl  cc15
     C599   60            9944            rts                ; table exhausted
                          9945
     C59A   AE 0279       9946    cc15    ldx  f2ptr
     C59D   B5 E7         9947            lda  pattyp,x
     C59F   30 05         9948            bmi  cc20
     C5A1   BD 0280       9949            lda  filtrk,x
     C5A4   D0 EE         9950            bne  cc10
     C5A6   A9 00         9951    cc20    lda  #0
     C5A8   8D 028F       9952            sta  found
     C5AB   60            9953            rts
```

```
Error Addr  Code           Seq    Source statement

                           9955
                           9956   ;  search directory
                           9957   ;  returns with valid entry w/ delind=0
                           9958   ;  or returns w/ 1st deleted entry
                           9959   ;  w/ delind=1
                           9960   ;
                           9961   ;  srchst will initiate a search
                           9962   ;  search will continue a search
                           9963
     C5AC                  9964   srchst
     C5AC   A0 00          9965          ldy  #0          ; init deleted sector
     C5AE   8C 0291        9966          sty  delsec
     C5B1   88             9967          dey
     C5B2   8C 0253        9968          sty  entfnd
                           9969
     C5B5   AD FE85        9970          lda  dirtrk      ; start search at beginning
     C5B8   85 80          9971          sta  track
     C5BA   A9 01          9972          lda  #1
     C5BC   85 81          9973          sta  sector
     C5BE   8D 0293        9974          sta  lstbuf
     C5C1   20 D475        9975          jsr  opnird      ; open internal read chnl
                           9976
     C5C4   AD 0293        9977   sr10   lda  lstbuf      ; last buffer if 0
     C5C7   D0 01          9978          bne  sr15
     C5C9   60             9979          rts              ; (z=1)
                           9980
     C5CA   A9 07          9981   sr15   lda  #7
     C5CC   8D 0295        9982          sta  filcnt
     C5CF   A9 00          9983          lda  #0          ; read track #
     C5D1   20 D4F6        9984          jsr  drdbyt
     C5D4   8D 0293        9985          sta  lstbuf      ; update end flag
                           9986
     C5D7   20 D4E8        9987   sr20   jsr  getpnt
     C5DA   CE 0295        9988          dec  filcnt
     C5DD   A0 00          9989          ldy  #0
     C5DF   B1 94          9990          lda  (dirbuf),y  ; read file type
     C5E1   D0 18          9991          bne  sr30
                           9992
     C5E3   AD 0291        9993          lda  delsec      ; deleted entry found
     C5E6   D0 2F          9994          bne  search      ; deleted entry already found
     C5E8   20 DE3B        9995          jsr  curblk      ; get current sector
     C5EB   A5 81          9996          lda  sector
     C5ED   8D 0291        9997          sta  delsec
                           9998
     C5F0   A5 94          9999          lda  dirbuf      ; get current index
     C5F2   AE 0292       10000          ldx  delind      ; bit1: want deleted entry
     C5F5   8D 0292       10001          sta  delind
     C5F8   F0 1D         10002          beq  search      ; need valid entry
     C5FA   60            10003          rts              ; (z=0)
                          10004
```

```
Error Addr  Code           Seq    Source statement

     C5FB   A2 01        10005   sr30    ldx  #1
     C5FD   EC 0292      10006           cpx  delind     ; ?looking for deleted?
     C600   D0 2D        10007           bne  sr50       ;  no!
     C602   F0 13        10008           beq  search
                         10009
     C604   AD FE85      10010   srre    lda  dirtrk
     C607   85 80        10011           sta  track
     C609   AD 0290      10012           lda  dirsec
     C60C   85 81        10013           sta. sector
     C60E   20 D475      10014           jsr  opnird
     C611   AD 0294      10015           lda  index
     C614   20 D4C8      10016           jsr  setpnt
                         10017
     C617   A9 FF        10018   search  lda  #$ff
     C619   8D 0253      10019           sta  entfnd
     C61C   AD 0295      10020           lda  filcnt     ; adjust file count
     C61F   30 08        10021           bmi  sr40
     C621   A9 20        10022           lda  #32        ; incr by 32
     C623   20 D1C6      10023           jsr  incptr
     C626   4C C5D7      10024           jmp  sr20
                         10025
     C629   20 D44D      10026   sr40    jsr  nxtbuf     ; new buffer
     C62C   4C C5C4      10027           jmp  sr10       ; (branch)
                         10028
     C62F   A5 94        10029   sr50    lda  dirbuf     ; found valid entry
     C631   8D 0294      10030           sta  index      ; save index
     C634   20 DE3B      10031           jsr  curblk     ; get sector
     C637   A5 81        10032           lda  sector
     C639   8D 0290      10033           sta  dirsec
                         10034
     C63C   60           10035           rts             ; (z=0)
     C63D                10036   auto1
                         10037   ;  check drive for active diskette
                         10038   ;  init if needed
                         10039   ;  return nodrv status
                         10040
     C63D   A5 68        10041           lda  autofg
     C63F   D0 28        10042           bne  auto2      ; auto-init is disabled
                         10043
     C641   A6 7F        10044           ldx  drvnum
     C643   56 1C        10045           lsr  wpsw,x     ; test & clear wpsw
     C645   90 22        10046           bcc  auto2      ; no change in diskette
                         10047
     C647   A9 FF        10048           lda  #$ff
     C649   8D 0298      10049           sta  jobrtn     ; set error return code
     C64C   20 D00E      10050           jsr  itrial     ; init-seek test
     C64F   A0 FF        10051           ldy  #$ff       ;  .y= true
     C651   C9 02        10052           cmp  #2
     C653   F0 0A        10053           beq  auto1      ; no sync= no diskette
                         10054
```

```
Error Addr  Code           Seq    Source statement

     C655   C9 03        10055            cmp  #3
     C657   F0 06        10056            beq  auto1      ; no header= no directory
                         10057
     C659   C9 0F        10058            cmp  #$f
     C65B   F0 02        10059            beq  auto1      ; no drive!!!!
                         10060
     C65D   A0 00        10061            ldy  #0         ; set .y false
     C65F                10062   auto1
     C65F   A6 7F        10063            ldx  drvnum
                         10064
                         10065   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         10066
     C661   4C A9A5      10067            jmp  ptch44     ; *** rom ds 05/01/85 ***
                         10068   ;        tya
                         10069   ;        sta  nodrv,x    ; set condn of no-drive
     C664                10070   rtch44                   ; ret address
                         10071
                         10072   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         10073
     C664   D0 03        10074            bne  auto2      ; no need to init crud!
                         10075
     C666   20 D042      10076            jsr  initdr     ; init that drive
     C669                10077   auto2
     C669   A6 7F        10078            ldx  drvnum
                         10079
                         10080   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         10081
     C66B   4C AA17      10082            jmp  ptch50     ; *** rom ds 05/01/85 ***
                         10083   ;        lda  nodrv,x    ; return no-drive condn
                         10084   ;        rts
                         10085
                         10086   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         10087            .include  trnsfr
```

```
Error Addr  Code           Seq   Source statement

                          10089          .subttl 'trnsfr.src'
                          10090
                          10091  ; transfer filename from cmd to buffer
                          10092  ; a: string size
                          10093  ; x: starting index in cmdbuf
                          10094  ; y: buffer #
     C66E   48            10095  trname  pha
     C66F   20 C6A6       10096          jsr  fndlmt
     C672   20 C688       10097          jsr  trcmbf
     C675   68            10098          pla
     C676   38            10099          sec
     C677   ED 024B       10100          sbc  strsiz
     C67A   AA            10101          tax
     C67B   F0 0A         10102          beq  tn20
     C67D   90 08         10103          bcc  tn20
                          10104
     C67F   A9 A0         10105          lda  #$a0
     C681   91 94         10106  tn10    sta  (dirbuf),y
     C683   C8            10107          iny
     C684   CA            10108          dex
     C685   D0 FA         10109          bne  tn10
     C687   60            10110  tn20    rts
                          10111
                          10112  ; transfer cmd buffer to other buffer
                          10113  ; uses current buffer ptr
                          10114  ; limit: ending index+1 in cmd buf
                          10115  ; x: starting index in cmd buf
                          10116  ; y: buffer #
                          10117
     C688   98            10118  trcmbf  tya
     C689   0A            10119          asl  a
     C68A   A8            10120          tay
     C68B   B9 0099       10121          lda  buftab,y
     C68E   85 94         10122          sta  dirbuf
     C690   B9 009A       10123          lda  buftab+1,y
     C693   85 95         10124          sta  dirbuf+1
     C695   A0 00         10125          ldy  #0
                          10126
     C697   BD 0200       10127  tr10    lda  cmdbuf,x
     C69A   91 94         10128          sta  (dirbuf),y
     C69C   C8            10129          iny
     C69D   F0 06         10130          beq  tr20
     C69F   E8            10131          inx
     C6A0   EC 0276       10132          cpx  limit
     C6A3   90 F2         10133          bcc  tr10
     C6A5   60            10134  tr20    rts
                          10135
                          10136
                          10137  ; find the limit of the string in cmdbuf
                          10138  ; pointed to by x
```

```
Error Addr  Code           Seq    Source statement

                          10139
      C6A6   A9 00        10140   fndlmt  lda  #0
      C6A8   8D 024B      10141           sta  strsiz
      C6AB   8A           10142           txa
      C6AC   48           10143           pha
      C6AD                10144   fl05
      C6AD   BD 0200      10145           lda  cmdbuf,x
      C6B0   C9 2C        10146           cmp  #','
      C6B2   F0 14        10147           beq  fl10
      C6B4   C9 3D        10148           cmp  #'=
      C6B6   F0 10        10149           beq  fl10
      C6B8   EE 024B      10150           inc  strsiz
      C6BB   E8           10151           inx
      C6BC   A9 0F        10152           lda  #15
      C6BE   CD 024B      10153           cmp  strsiz
      C6C1   90 05        10154           bcc  fl10
      C6C3   EC 0274      10155           cpx  cmdsiz
      C6C6   90 E5        10156           bcc  fl05
      C6C8   8E 0276      10157   fl10    stx  limit
      C6CB   68           10158           pla
      C6CC   AA           10159           tax
      C6CD   60           10160           rts
                          10161
```

```
Error Addr  Code           Seq    Source statement

                          10163   ; get file entry from directory
                          10164   ; called by stdir, getdir
                          10165
     C6CE   A5 83         10166   getnam  lda  sa          ; save variables
     C6D0   48            10167           pha
     C6D1   A5 82         10168           lda  lindx
     C6D3   48            10169           pha
     C6D4   20 C6DE       10170           jsr  gnsub
     C6D7   68            10171           pla             ; restore variables
     C6D8   85 82         10172           sta  lindx
     C6DA   68            10173           pla
     C6DB   85 83         10174           sta  sa
     C6DD   60            10175           rts
                          10176
     C6DE   A9 11         10177   gnsub   lda  #irsa
     C6E0   85 83         10178           sta  sa
     C6E2   20 D0EB       10179           jsr  fndrch
     C6E5   20 D4E8       10180           jsr  getpnt
     C6E8   AD 0253       10181           lda  entfnd
     C6EB   10 0A         10182           bpl  gn05       ; more files
     C6ED   AD 028D       10183           lda  drvflg
     C6F0   D0 0A         10184           bne  gn050
     C6F2   20 C806       10185           jsr  msgfre     ; send blocks free
     C6F5   18            10186           clc             ; (c=0): end
     C6F6   60            10187           rts             ; terminate
                          10188
     C6F7   AD 028D       10189   gn05    lda  drvflg     ; (drvflg=0):
     C6FA   F0 1F         10190           beq  gn10       ;  send file name
     C6FC                 10191   gn050
     C6FC   CE 028D       10192           dec  drvflg     ; (drvflg=-1):new dir
     C6FF   D0 0D         10193           bne  gn051
     C701   CE 028D       10194           dec  drvflg     ; (drvflg=-1):
     C704   20 C38F       10195           jsr  togdrv     ;  send blocks free
     C707   20 C806       10196           jsr  msgfre
     C70A   38            10197           sec
     C70B   4C C38F       10198           jmp  togdrv
                          10199
     C70E   A9 00         10200   gn051   lda  #0
     C710   8D 0273       10201           sta  nbtemp+1
     C713   8D 028D       10202           sta  drvflg     ; reset flag
     C716   20 C7B7       10203           jsr  newdir
     C719   38            10204           sec
     C71A   60            10205           rts
                          10206
     C71B   A2 18         10207   gn10    ldx  #dirlen    ; set number blocks
     C71D   A0 1D         10208           ldy  #29        ;  & adjust spacing
     C71F   B1 94         10209           lda  (dirbuf),y
     C721   8D 0273       10210           sta  nbtemp+1
     C724   F0 02         10211           beq  gn12
     C726   A2 16         10212           ldx  #dirlen-2
```

```
Error Addr  Code           Seq    Source statement

     C728   88            10213  gn12    dey
     C729   B1 94         10214          lda  (dirbuf),y
     C72B   8D 0272       10215          sta  nbtemp
     C72E   E0 16         10216          cpx  #dirlen-2
     C730   F0 0A         10217          beq  gn14
     C732   C9 0A         10218          cmp  #10
     C734   90 06         10219          bcc  gn14
     C736   CA            10220          dex
     C737   C9 64         10221          cmp  #100
     C739   90 01         10222          bcc  gn14
     C73B   CA            10223          dex
                          10224
     C73C   20 C7AC       10225  gn14    jsr  blknb      ; clear name buffer
                          10226
     C73F   B1 94         10227          lda  (dirbuf),y ; set type chars
     C741   48            10228          pha
     C742   0A            10229          asl  a          ; (used in bcs)
     C743   10 05         10230          bpl  gn15
     C745   A9 3C         10231          lda  #'<
     C747   9D 02B2       10232          sta  nambuf+1,x
     C74A                 10233  gn15
     C74A   68            10234          pla
     C74B   29 0F         10235          and  #$f
     C74D   A8            10236          tay
     C74E   B9 FEC5       10237          lda  tp2lst,y
     C751   9D 02B1       10238          sta  nambuf,x
     C754   CA            10239          dex
     C755   B9 FEC0       10240          lda  tp1lst,y
     C758   9D 02B1       10241          sta  nambuf,x
     C75B   CA            10242          dex
     C75C   B9 FEBB       10243          lda  typlst,y
     C75F   9D 02B1       10244          sta  nambuf,x
     C762   CA            10245          dex
     C763   CA            10246          dex
     C764   B0 05         10247          bcs  gn20       ; (from asl)
     C766   A9 2A         10248          lda  #'*        ; file not closed
     C768   9D 02B2       10249          sta  nambuf+1,x
     C76B   A9 A0         10250  gn20    lda  #$a0
     C76D   9D 02B1       10251          sta  nambuf,x
     C770   CA            10252          dex
     C771   A0 12         10253          ldy  #18
     C773   B1 94         10254  gn22    lda  (dirbuf),y
     C775   9D 02B1       10255          sta  nambuf,x
     C778   CA            10256          dex
     C779   88            10257          dey
     C77A   C0 03         10258          cpy  #3
     C77C   B0 F5         10259          bcs  gn22
                          10260
     C77E   A9 22         10261          lda  #'"        ; send name in quotes
     C780   9D 02B1       10262          sta  nambuf,x
```

```
Error Addr  Code           Seq    Source statement

     C783   E8           10263   gn30    inx
     C784   E0 20        10264           cpx  #$20
     C786   B0 0B        10265           bcs  gn35
     C788   BD 02B1      10266           lda  nambuf,x
     C78B   C9 22        10267           cmp  #'"
     C78D   F0 04        10268           beq  gn35
     C78F   C9 A0        10269           cmp  #$a0
     C791   D0 F0        10270           bne  gn30
     C793   A9 22        10271   gn35    lda  #'"
     C795   9D 02B1      10272           sta  nambuf,x
     C798   E8           10273   gn37    inx
     C799   E0 20        10274           cpx  #$20
     C79B   B0 0A        10275           bcs  gn40
     C79D   A9 7F        10276           lda  #$7f
     C79F   3D 02B1      10277           and  nambuf,x
     C7A2   9D 02B1      10278           sta  nambuf,x
     C7A5   10 F1        10279           bpl  gn37
                         10280
     C7A7   20 C4B5      10281   gn40    jsr  fndfil
     C7AA   38           10282           sec
     C7AB   60           10283   gn45    rts
                         10284
     C7AC   A0 1B        10285   blknb   ldy  #nbsiz      ; blank nambuf
     C7AE   A9 20        10286           lda  #$20
     C7B0   99 02B0      10287   blknb1  sta  nambuf-1,y
     C7B3   88           10288           dey
     C7B4   D0 FA        10289           bne  blknb1
     C7B6   60           10290           rts
                         10291
                         10292   ;new directory in listing
                         10293
     C7B7                10294   newdir
     C7B7   20 F119      10295           jsr  bam2x
     C7BA   20 F0DF      10296           jsr  redbam
     C7BD   20 C7AC      10297           jsr  blknb
     C7C0   A9 FF        10298           lda  #$ff
     C7C2   85 6F        10299           sta  temp
     C7C4   A6 7F        10300           ldx  drvnum
     C7C6   8E 0272      10301           stx  nbtemp
     C7C9   A9 00        10302           lda  #0
     C7CB   8D 0273      10303           sta  nbtemp+1
     C7CE   A6 F9        10304           ldx  jobnum
     C7D0   BD FEE0      10305           lda  bufind,x
     C7D3   85 95        10306           sta  dirbuf+1
     C7D5   AD FE88      10307           lda  dsknam
     C7D8   85 94        10308           sta  dirbuf
     C7DA   A0 16        10309           ldy  #22
                         10310
     C7DC   B1 94        10311   nd10    lda  (dirbuf),y
     C7DE   C9 A0        10312           cmp  #$a0
```

```
Error Addr  Code           Seq    Source statement

                          10347           .subttl 'scrtch.src'
                          10348
                          10349   ; scratch file(s)
     C823   20 C398       10350   scrtch  jsr  fs1set     ; set up for 1 stream
     C826   20 C320       10351           jsr  alldrs
     C829   20 C3CA       10352           jsr  optsch
     C82C   A9 00         10353           lda  #0
     C82E   85 86         10354           sta  r0          ; used as file count
     C830   20 C49D       10355           jsr  ffst
     C833   30 3D         10356           bmi  sc30
     C835   20 DDB7       10357   sc15    jsr  tstchn      ; is it active ?
     C838   90 33         10358           bcc  sc25        ; yes - dont scratch
     C83A   A0 00         10359           ldy  #0
     C83C   B1 94         10360           lda  (dirbuf),y
     C83E   29 40         10361           and  #$40        ; lock bit
     C840   D0 2B         10362           bne  sc25        ; it's locked
                          10363
     C842   20 C8B6       10364           jsr  deldir      ; delete directory
     C845   A0 13         10365           ldy  #19         ; is this a relative ?
     C847   B1 94         10366           lda  (dirbuf),y  ; has a ss ?
     C849   F0 0A         10367           beq  sc17        ; no
     C84B   85 80         10368           sta  track       ; yes - save track
     C84D   C8            10369           iny
     C84E   B1 94         10370           lda  (dirbuf),y  ; get sector
     C850   85 81         10371           sta  sector
     C852   20 C87D       10372           jsr  delfil      ; delete by links
     C855   AE 0253       10373   sc17    ldx  entfnd
     C858   A9 20         10374           lda  #$20
     C85A   35 E7         10375           and  pattyp,x
     C85C   D0 0D         10376           bne  sc20        ; created, not closed
                          10377
     C85E   BD 0280       10378           lda  filtrk,x    ; delete by links
     C861   85 80         10379           sta  track
     C863   BD 0285       10380           lda  filsec,x
     C866   85 81         10381           sta  sector
     C868   20 C87D       10382           jsr  delfil
                          10383
     C86B   E6 86         10384   sc20    inc  r0
     C86D   20 C48B       10385   sc25    jsr  ffre
     C870   10 C3         10386           bpl  sc15
                          10387
     C872   A5 86         10388   sc30    lda  r0          ; finished, set
     C874   85 80         10389           sta  track       ;  file count
     C876   A9 01         10390           lda  #1
     C878   A0 00         10391           ldy  #0
     C87A   4C C1A3       10392           jmp  scrend      ; end of scratch
                          10393
     C87D   20 EF5F       10394   delfil  jsr  frets       ; delete file by links
     C880   20 D475       10395           jsr  opnird      ; update bam
     C883   20 F119       10396           jsr  bam2x
```

```
Error Addr  Code           Seq    Source statement

     C7E0   D0 0B        10313            bne  nd20
     C7E2   A9 31        10314            lda  #'1         ; version # 1
     C7E4   2C           10315            .byte skip2
     C7E5                10316    nd15
     C7E5   B1 94        10317            lda  (dirbuf),y
     C7E7   C9 A0        10318            cmp  #$a0
     C7E9   D0 02        10319            bne  nd20
                         10320
     C7EB   A9 20        10321            lda  #$20
     C7ED                10322    nd20
     C7ED   99 02B3      10323            sta  nambuf+2,y
     C7F0   88           10324            dey
     C7F1   10 F2        10325            bpl  nd15
     C7F3   A9 12        10326            lda  #$12
     C7F5   8D 02B1      10327            sta  nambuf
     C7F8   A9 22        10328            lda  #'"
     C7FA   8D 02B2      10329            sta  nambuf+1
     C7FD   8D 02C3      10330            sta  nambuf+18
     C800   A9 20        10331            lda  #$20
     C802   8D 02C4      10332            sta  nambuf+19
     C805   60           10333            rts
                         10334
     C806   20 C7AC      10335    msgfre  jsr  blknb
     C809   A0 0B        10336            ldy  #msglen-1
     C80B   B9 C817      10337    msg1    lda  fremsg,y
     C80E   99 02B1      10338            sta  nambuf,y
     C811   88           10339            dey
     C812   10 F7        10340            bpl  msg1
     C814   4C EF4D      10341            jmp  numfre
                         10342
     C817   42 4C 4F     10343    fremsg   .byte  'BLOCKS FREE.'
     C81A   43 4B 53
     C81D   20 46 52
     C820   45 45 2E
            =000C        10344    msglen   =*-fremsg
                         10345            .include  scrtch
```

```
Error Addr  Code           Seq    Source statement

     C886   B5 A7        10397            lda  buf0,x
     C888   C9 FF        10398            cmp  #$ff
     C88A   F0 08        10399            beq  del10
     C88C   AD 02F9      10400            lda  wbam
     C88F   09 40        10401            ora  #$40
     C891   8D 02F9      10402            sta  wbam
     C894                10403   del10
     C894   A9 00        10404   del2     lda  #0
     C896   20 D4C8      10405            jsr  setpnt
     C899   20 D156      10406            jsr  rdbyt
     C89C   85 80        10407            sta  track
     C89E   20 D156      10408            jsr  rdbyt
     C8A1   85 81        10409            sta  sector
     C8A3   A5 80        10410            lda  track
     C8A5   D0 06        10411            bne  del1
     C8A7   20 EEF4      10412            jsr  mapout
     C8AA   4C D227      10413            jmp  frechn
     C8AD   20 EF5F      10414   del1     jsr  frets
     C8B0   20 D44D      10415            jsr  nxtbuf
     C8B3   4C C894      10416            jmp  del2
                         10417
     C8B6   A0 00        10418   deldir   ldy  #0          ; delete dir entry
     C8B8   98           10419            tya
     C8B9   91 94        10420            sta  (dirbuf),y
     C8BB   20 DE5E      10421            jsr  wrtout
     C8BE   4C D599      10422            jmp  watjob
                         10423            .include  duplct
```

```
Error Addr  Code           Seq    Source statement

                          10425           .subttl 'duplct.src'
                          10426
                          10427   ; duplicate disk
                          10428
     C8C1                 10429   duplct
     C8C1   A9 31         10430           lda  #badcmd
     C8C3   4C C1C8       10431           jmp  cmderr
                          10432
                          10433   ; transfer format code to buffer 0
                          10434   ;  & start controller formatting
                          10435
     C8C6                 10436   format
     C8C6   A9 4C         10437           lda  #$4c
     C8C8   8D 0600       10438           sta  bufs+$300
                          10439
     C8CB   A9 C7         10440           lda  #<formt
     C8CD   8D 0601       10441           sta  bufs+$301
     C8D0   A9 FA         10442           lda  #>formt
     C8D2   8D 0602       10443           sta  bufs+$302
                          10444
     C8D5   A9 03         10445           lda  #3
     C8D7   20 D6D3       10446           jsr  seth
     C8DA   A5 7F         10447           lda  drvnum
     C8DC   09 E0         10448           ora  #exec
     C8DE   85 03         10449           sta  jobs+3
     C8E0   A5 03         10450   fmt105  lda  jobs+3
     C8E2   30 FC         10451           bmi  fmt105
                          10452
     C8E4   C9 02         10453           cmp  #2
     C8E6   90 07         10454           bcc  fmt110
                          10455
     C8E8   A9 03         10456           lda  #3
     C8EA   A2 00         10457           ldx  #0
     C8EC   4C E60A       10458           jmp  error
     C8EF   60            10459   fmt110  rts
                          10460           .include  copset
```

```
Error Addr  Code           Seq    Source statement

                          10462           .subttl 'copset.src'
                          10463
                          10464   ; dskcpy check for type
                          10465   ; and parses special case
                          10466
     C8F0                 10467   dskcpy
     C8F0   A9 E0         10468           lda  #$e0       ; kill bam buffer
     C8F2   8D 024F       10469           sta  bufuse
     C8F5   20 FOD1       10470           jsr  clnbam     ; clr tbam
     C8F8   20 F119       10471           jsr  bam2x      ; get bam lindx in .x
     C8FB   A9 FF         10472           lda  #$ff
     C8FD   95 A7         10473           sta  buf0,x     ; mark bam out-of-memory
     C8FF   A9 OF         10474           lda  #$0f
     C901   8D 0256       10475           sta  linuse     ; free all lindxs
     C904   20 C1E5       10476           jsr  prscln     ; find ":"
     C907   DO 03         10477           bne  dx0000
     C909   4C C8C1       10478           jmp  duplct     ; bad command error, cx=x not allowed
     C90C   20 C1F8       10496   dx0000  jsr  tc30       ; normal parse
     C90F   20 C320       10497   dx0005  jsr  alldrs     ; put drv's in filtbl
     C912   AD 028B       10498           lda  image      ; get parse image
     C915   29 55         10499           and  #%01010101 ; val for patt copy
     C917   DO OF         10500           bne  dx0020     ; must be concat or normal
     C919   AE 027A       10501           ldx  filtbl     ; chk for *
     C91C   BD 0200       10502           lda  cmdbuf,x
     C91F   C9 2A         10503           cmp  #'*
     C921   DO 05         10504           bne  dx0020
     C923   A9 30         10513   dx0010  lda  #badsyn    ; syntax error
     C925   4C C1C8       10514           jmp  cmderr
     C928   AD 028B       10515   dx0020  lda  image      ; chk for normal
     C92B   29 D9         10516           and  #%11011001
     C92D   DO F4         10517           bne  dx0010
     C92F   4C C952       10518           jmp  copy
                          10541           .include  copall
```

```
Error Addr  Code           Seq    Source statement

                          10543           .subttl 'copall.src'
                          10544
                          10545   ; set up subroutine
                          10546
     C932   A9 00         10547   pups1   lda  #0
     C934   8D 0258       10548           sta  rec
     C937   8D 028C       10549           sta  drvcnt
     C93A   8D 0280       10550           sta  filtrk
     C93D   8D 0281       10551           sta  filtrk+1
     C940   A5 E3         10552           lda  fildrv+1   ; get drive number
     C942   29 01         10553           and  #1         ; only
     C944   85 7F         10554           sta  drvnum
     C946   09 01         10555           ora  #1
     C948   8D 0291       10556           sta  delsec     ; nonzero
     C94B   AD 027B       10557           lda  filtbl+1   ; fn1=fn2
     C94E   8D 027A       10558           sta  filtbl
     C951   60            10559           rts
                          10646           .include  rename
```

```
Error Addr  Code           Seq    Source statement

                          10648           .subttl 'rename.src'
                          10649
                          10650   ; copy file(s) to one file
                          10651
     C952                 10652   copy                    ; filenames, optimize
     C952   20 C44F       10653           jsr  lookup     ; look ip all files
     C955   AD 0278       10654           lda  f2cnt
     C958   C9 03         10655           cmp  #3
     C95A   90 45         10656           bcc  cop10
                          10657
     C95C   A5 E2         10658           lda  fildrv
     C95E   C5 E3         10659           cmp  fildrv+1
     C960   D0 3F         10660           bne  cop10
                          10661
     C962   A5 DD         10662           lda  entind
     C964   C5 DE         10663           cmp  entind+1
     C966   D0 39         10664           bne  cop10
                          10665
     C968   A5 D8         10666           lda  entsec
     C96A   C5 D9         10667           cmp  entsec+1
     C96C   D0 33         10668           bne  cop10
                          10669
     C96E   20 CACC       10670           jsr  chkin      ; concat
     C971   A9 01         10671           lda  #1
     C973   8D 0279       10672           sta  f2ptr
     C976   20 C9FA       10673           jsr  opirfl
                          10674
     C979   20 D125       10675           jsr  typfil
     C97C   F0 04         10676           beq  cop01
     C97E   C9 02         10677           cmp  #prgtyp
     C980   D0 05         10678           bne  cop05
     C982                 10679   cop01
     C982   A9 64         10680           lda  #mistyp
     C984   20 C1C8       10681           jsr  cmderr
     C987                 10682   cop05
     C987   A9 12         10683           lda  #iwsa
     C989   85 83         10684           sta  sa
     C98B   AD 023C       10685           lda  lintab+irsa
     C98E   8D 023D       10686           sta  lintab+iwsa
     C991   A9 FF         10687           lda  #$ff
     C993   8D 023C       10688           sta  lintab+irsa
     C996   20 DA2A       10689           jsr  append
     C999   A2 02         10690           ldx  #2
     C99B   20 C9B9       10691           jsr  cy10
     C99E   4C C194       10692           jmp  endcmd
     C9A1                 10693   cop10
     C9A1   20 C9A7       10694           jsr  cy
     C9A4   4C C194       10695           jmp  endcmd
                          10696
                          10697
```

```
Error Addr  Code           Seq    Source statement

     C9A7                10698   cy
     C9A7   20 CAE7      10699           jsr  chkio      ; check files for existence
                         10700
     C9AA   A5 E2        10701           lda  fildrv
     C9AC   29 01        10702           and  #1
     C9AE   85 7F        10703           sta  drvnum
     C9B0   20 D486      10704           jsr  opniwr     ; open internal write chnl
     C9B3   20 D6E4      10705           jsr  addfil     ; add to directory
     C9B6   AE 0277      10706           ldx  ficnt
                         10707
     C9B9   8E 0279      10708   cy10    stx  f2ptr      ; set up read file
     C9BC   20 C9FA      10709           jsr  opirfl
                         10710
     C9BF   A9 11        10711           lda  #irsa      ; add for rel copy
     C9C1   85 83        10712           sta  sa
     C9C3   20 D0EB      10713           jsr  fndrch
     C9C6   20 D125      10714           jsr  typfil
     C9C9   D0 03        10715           bne  cy10a      ; not rel
     C9CB   20 CA53      10716           jsr  cyext
                         10717
     C9CE   A9 08        10718   cy10a   lda  #eoisnd
     C9D0   85 F8        10719           sta  eoiflg
     C9D2   4C C9D8      10720           jmp  cy20
     C9D5                10721   cy15
     C9D5   20 CF9B      10722           jsr  pibyte
     C9D8                10723   cy20
     C9D8   20 CA35      10724           jsr  gibyte
     C9DB   A9 80        10725           lda  #lrf
     C9DD   20 DDA6      10726           jsr  tstflg
     C9E0   F0 F3        10727           beq  cy15
                         10728
     C9E2   20 D125      10729           jsr  typfil
     C9E5   F0 03        10730           beq  cy30
                         10731
     C9E7   20 CF9B      10732           jsr  pibyte
     C9EA                10733   cy30
     C9EA   AE 0279      10734           ldx  f2ptr
                         10735
     C9ED   E8           10736           inx
     C9EE   EC 0278      10737           cpx  f2cnt
     C9F1   90 C6        10738           bcc  cy10       ; more files to copy
     C9F3   A9 12        10739           lda  #iwsa
     C9F5   85 83        10740           sta  sa
     C9F7   4C DB02      10741           jmp  clschn     ; close copy channel, file
                         10742
     C9FA                10743   opirfl
     C9FA   AE 0279      10744           ldx  f2ptr
     C9FD   B5 E2        10745           lda  fildrv,x
     C9FF   29 01        10746           and  #1
     CA01   85 7F        10747           sta  drvnum
```

```
Error Addr  Code           Seq    Source statement

     CA03   AD FE85      10748            lda  dirtrk
     CA06   85 80        10749            sta  track
     CA08   B5 D8        10750            lda  entsec,x
     CA0A   85 81        10751            sta  sector
     CA0C   20 D475      10752            jsr  opnird
     CA0F   AE 0279      10753            ldx  f2ptr
     CA12   B5 DD        10754            lda  entind,x
     CA14   20 D4C8      10755            jsr  setpnt
     CA17   AE 0279      10756            ldx  f2ptr
     CA1A   B5 E7        10757            lda  pattyp,x
     CA1C   29 07        10758            and  #typmsk
     CA1E   8D 024A      10759            sta  type
                         10760
     CA21   A9 00        10761            lda  #0
     CA23   8D 0258      10762            sta  rec
     CA26   20 D9A0      10763            jsr  opread
     CA29   A0 01        10764            ldy  #1
     CA2B   20 D125      10765            jsr  typfil
     CA2E   F0 01        10766            beq  opir10
     CA30   C8           10767            iny
     CA31                10768  opir10
     CA31   98           10769            tya
     CA32   4C D4C8      10770            jmp  setpnt
                         10771
     CA35                10772  gibyte
     CA35   A9 11        10773            lda  #irsa
     CA37   85 83        10774            sta  sa
     CA39                10775  gcbyte
     CA39   20 D39B      10776            jsr  gbyte
                         10777
     CA3C   85 85        10778            sta  data
     CA3E   A6 82        10779            ldx  lindx
     CA40   B5 F2        10780            lda  chnrdy,x
     CA42   29 08        10781            and  #eoisnd
     CA44   85 F8        10782            sta  eoiflg
     CA46   D0 0A        10783            bne  gib20
                         10784
     CA48   20 D125      10785            jsr  typfil
     CA4B   F0 05        10786            beq  gib20
                         10787
     CA4D   A9 80        10788            lda  #lrf
     CA4F   20 DD97      10789            jsr  setflg
     CA52   60           10790  gib20     rts
                         10791
     CA53   20 D1D3      10792  cyext     jsr  setdrn      ; copy rel records
     CA56   20 E1CB      10793            jsr  ssend
     CA59   A5 D6        10794            lda  ssind
     CA5B   48           10795            pha
     CA5C   A5 D5        10796            lda  ssnum
     CA5E   48           10797            pha
```

```
Error Addr  Code           Seq    Source statement

     CA5F   A9 12        10798            lda  #iwsa
     CA61   85 83        10799            sta  sa
     CA63   20 D107      10800            jsr  fndwch
     CA66   20 D1D3      10801            jsr  setdrn
     CA69   20 E1CB      10802            jsr  ssend
     CA6C   20 E29C      10803            jsr  posbuf
     CA6F   A5 D6        10804            lda  ssind
     CA71   85 87        10805            sta  r1
     CA73   A5 D5        10806            lda  ssnum
     CA75   85 86        10807            sta  r0
     CA77   A9 00        10808            lda  #0
     CA79   85 88        10809            sta  r2
     CA7B   85 D4        10810            sta  recptr
     CA7D   85 D7        10811            sta  relptr
     CA7F   68           10812            pla
     CA80   85 D5        10813            sta  ssnum
     CA82   68           10814            pla
     CA83   85 D6        10815            sta  ssind
     CA85   4C E33B      10816            jmp  addr1
                         10817
```

```
Error Addr  Code           Seq    Source statement

                          10819
                          10820  ; rename file name in directory
                          10821
     CA88   20 C320       10822  rename  jsr  alldrs     ; set both drive #'s
     CA8B   A5 E3         10823          lda  fildrv+1
     CA8D   29 01         10824          and  #1
     CA8F   85 E3         10825          sta  fildrv+1
     CA91   C5 E2         10826          cmp  fildrv
     CA93   F0 02         10827          beq  rn10       ; same drive #'s
     CA95   09 80         10828          ora  #$80       ; check both drives for name
     CA97   85 E2         10829  rn10    sta  fildrv
     CA99   20 C44F       10830          jsr  lookup     ; look up both names
     CA9C   20 CAE7       10831          jsr  chkio      ; check for existence
     CA9F   A5 E3         10832          lda  fildrv+1
     CAA1   29 01         10833          and  #1
     CAA3   85 7F         10834          sta  drvnum
     CAA5   A5 D9         10835          lda  entsec+1
     CAA7   85 81         10836          sta  sector
     CAA9   20 DE57       10837          jsr  rdab       ; read directory sector
     CAAC   20 D599       10838          jsr  watjob
     CAAF   A5 DE         10839          lda  entind+1
     CAB1   18            10840          clc             ; set sector index
     CAB2   69 03         10841          adc  #3         ; ...+3
     CAB4   20 D4C8       10842          jsr  setpnt
     CAB7   20 DF93       10843          jsr  getact
     CABA   A8            10844          tay
     CABB   AE 027A       10845          ldx  filtbl
     CABE   A9 10         10846          lda  #16
     CAC0   20 C66E       10847          jsr  trname     ; transfer name
     CAC3   20 DE5E       10848          jsr  wrtout     ; write sector out
     CAC6   20 D599       10849          jsr  watjob
     CAC9   4C C194       10850          jmp  endcmd
                          10851
                          10852
                          10853  ; check i/o file for exist
                          10854
     CACC                 10855  chkin
     CACC   A5 E8         10856          lda  pattyp+1   ; 1st file bears type
     CACE   29 07         10857          and  #typmsk
     CAD0   8D 024A       10858          sta  type
                          10859
     CAD3   AE 0278       10860          ldx  f2cnt
     CAD6   CA            10861  ck10    dex
     CAD7   EC 0277       10862          cpx  f1cnt
     CADA   90 0A         10863          bcc  ck20
     CADC   BD 0280       10864          lda  filtrk,x
     CADF   D0 F5         10865          bne  ck10
     CAE1   A9 62         10866          lda  #flntfd    ; input file not found
     CAE3   4C C1C8       10867          jmp  cmderr
     CAE6                 10868  ck20
```

```
Error Addr  Code           Seq    Source statement

     CAE6   60           10869          rts
                        10870
     CAE7               10871    chkio
     CAE7   20 CACC      10872          jsr  chkin
     CAEA   BD 0280      10873    ck25  lda  filtrk,x
     CAED   F0 05        10874          beq  ck30
     CAEF   A9 63        10875          lda  #flexst
     CAF1   4C C1C8      10876          jmp  cmderr
     CAF4   CA           10877    ck30  dex
     CAF5   10 F3        10878          bpl  ck25
     CAF7   60           10879          rts
                        10880          .include  memrw
```

```
Error Addr  Code           Seq    Source statement

                          10882            .subttl 'memrw.src'
                          10883
                          10884    ;  memory access commands
                          10885    ;  "-" must be 2nd char
     CAF8   AD 0201       10886    mem      lda  cmdbuf+1
     CAFB   C9 2D         10887             cmp  #'-
     CAFD   D0 4C         10888             bne  memerr
                          10889
     CAFF   AD 0203       10890             lda  cmdbuf+3   ; set address in temp
     CB02   85 6F         10891             sta  temp
     CB04   AD 0204       10892             lda  cmdbuf+4
     CB07   85 70         10893             sta  temp+1
                          10894
     CB09   A0 00         10895             ldy  #0
     CB0B   AD 0202       10896             lda  cmdbuf+2
     CB0E   C9 52         10897             cmp  #'R
     CB10   F0 0E         10898             beq  memrd      ; read
     CB12   20 F258       10899             jsr  killp      ; kill protect
     CB15   C9 57         10900             cmp  #'W
     CB17   F0 37         10901             beq  memwrt     ; write
     CB19   C9 45         10902             cmp  #'E
     CB1B   D0 2E         10903             bne  memerr     ; error
                          10904
                          10905    ; execute
                          10906
     CB1D   6C 006F       10907    memex    jmp  (temp)
                          10908
     CB20                 10909    memrd
     CB20   B1 6F         10910             lda  (temp),y
     CB22   85 85         10911             sta  data
     CB24   AD 0274       10912             lda  cmdsiz
     CB27   C9 06         10913             cmp  #6
     CB29   90 1A         10914             bcc  m30
                          10915
     CB2B   AE 0205       10916             ldx  cmdbuf+5
     CB2E   CA            10917             dex
     CB2F   F0 14         10918             beq  m30
     CB31   8A            10919             txa
     CB32   18            10920             clc
     CB33   65 6F         10921             adc  temp
     CB35   E6 6F         10922             inc  temp
     CB37   8D 0249       10923             sta  lstchr+errchn
     CB3A   A5 6F         10924             lda  temp
     CB3C   85 A5         10925             sta  cb+2
     CB3E   A5 70         10926             lda  temp+1
     CB40   85 A6         10927             sta  cb+3
     CB42   4C D443       10928             jmp  ge20
     CB45                 10929    m30
     CB45   20 D0EB       10930             jsr  fndrch
     CB48   4C D43A       10931             jmp  ge15
```

```
Error Addr  Code           Seq    Source statement

                          10932
     CB4B   A9 31         10933   memerr  lda  #badcmd     ; bad command
     CB4D   4C C1C8       10934           jmp  cmderr
                          10935
     CB50                 10936   memwrt                   ; write
     CB50   B9 0206       10937   m10     lda  cmdbuf+6,y
     CB53   91 6F         10938           sta  (temp),y    ; transfer from cmdbuf
     CB55   C8            10939           iny
     CB56   CC 0205       10940           cpy  cmdbuf+5    ; # of bytes to write
     CB59   90 F5         10941           bcc  m10
     CB5B   60            10942           rts
                          10943           .include  block
```

```
Error Addr  Code           Seq    Source statement

                          10945           .subttl 'block.src'
                          10946
                          10947   ; rom 1.1 additions
                          10948   ; user commands
                          10949
     CB5C   AC 0201       10950   user    ldy  cmdbuf+1
     CB5F   C0 30         10951           cpy  #'0
     CB61   D0 09         10952           bne  us10       ; 0 resets pntr
                          10953
     CB63   4C 8030       10954   usrint  jmp  burst_routines
     CB66   EA            10955           nop             ; fill
     CB67   EA            10956           nop             ; fill
     CB68   EA            10957           nop             ; fill
     CB69   EA            10958           nop             ; fill
     CB6A   EA            10959           nop             ; fill
     CB6B   EA            10960           nop             ; fill
                          10961
                          10962   ;       lda  #<ublock   ; set default block add
                          10963   ;       sta  usrjmp
                          10964   ;       lda  #>ublock
                          10965   ;       sta  usrjmp+1
                          10966   ;       rts
                          10967
     CB6C   20 CB72       10968   us10    jsr  usrexc     ; execute code by table
     CB6F   4C C194       10969           jmp  endcmd
                          10970
     CB72   88            10971   usrexc  dey             ; entry is(((index-1)and$f)*2)
     CB73   98            10972           tya
     CB74   29 0F         10973           and  #$f
     CB76   0A            10974           asl  a
     CB77   A8            10975           tay
                          10976
     CB78   B1 6B         10977           lda  (usrjmp),y
     CB7A   85 75         10978           sta  ip
     CB7C   C8            10979           iny
     CB7D   B1 6B         10980           lda  (usrjmp),y
     CB7F   85 76         10981           sta  ip+1
                          10982
                          10983   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          10990   ;  open direct access buffer
                          10991   ;  from open "#"
                          10992
     CB84   AD 028E       10993   opnblk  lda  lstdrv
     CB87   85 7F         10994           sta  drvnum
     CB89   A5 83         10995           lda  sa          ; sa is destroyed by this patch
     CB8B   48            10996           pha
     CB8C   20 C63D       10997           jsr  autoi       ; init disk for proper channel assignment
     CB8F   68            10998           pla              ; restore sa
     CB90   85 83         10999           sta  sa
     CB92   AE 0274       11000           ldx  cmdsiz
     CB95   CA            11001           dex
     CB96   D0 0D         11002           bne  ob10
                          11003
     CB98   A9 01         11004           lda  #1          ; get any buffer
     CB9A   20 D1E2       11005           jsr  getrch
     CB9D   4C CBF1       11006           jmp  ob30
                          11007
     CBA0   A9 70         11008   ob05    lda  #nochnl
     CBA2   4C C1C8       11009           jmp  cmderr
                          11010
     CBA5   A0 01         11011   ob10    ldy  #1          ; buffer # is requested
     CBA7   20 CC7C       11012           jsr  bp05
     CBAA   AE 0285       11013           ldx  filsec
     CBAD   E0 05         11014           cpx  #bfcnt      ; must be less than 6.
     CBAF   B0 EF         11015           bcs  ob05
                          11016
     CBB1   A9 00         11017           lda  #0
     CBB3   85 6F         11018           sta  temp
     CBB5   85 70         11019           sta  temp+1
     CBB7   38            11020           sec
                          11021
     CBB8                 11022   ob15
     CBB8   26 6F         11023           rol  temp
     CBBA   26 70         11024           rol  temp+1
     CBBC   CA            11025           dex
     CBBD   10 F9         11026           bpl  ob15
                          11027
     CBBF   A5 6F         11028           lda  temp
     CBC1   2D 024F       11029           and  bufuse
     CBC4   D0 DA         11030           bne  ob05        ; buffer is used
     CBC6   A5 70         11031           lda  temp+1
     CBC8   2D 0250       11032           and  bufuse+1
     CBCB   D0 D3         11033           bne  ob05        ; buf is used
                          11034
     CBCD   A5 6F         11035           lda  temp
     CBCF   0D 024F       11036           ora  bufuse      ; set buffer as used
     CBD2   8D 024F       11037           sta  bufuse
     CBD5   A5 70         11038           lda  temp+1
     CBD7   0D 0250       11039           ora  bufuse+1
```

```
Error Addr  Code           Seq    Source statement

     CBDA   8D 0250      11040            sta  bufuse+1
                         11041
     CBDD   A9 00        11042            lda  #0          ; set up channel
     CBDF   20 D1E2      11043            jsr  getrch
     CBE2   A6 82        11044            ldx  lindx
     CBE4   AD 0285      11045            lda  filsec
     CBE7   95 A7        11046            sta  buf0,x
     CBE9   AA           11047            tax
     CBEA   A5 7F        11048            lda  drvnum
     CBEC   95 00        11049            sta  jobs,x
     CBEE   9D 025B      11050            sta  lstjob,x
                         11051
     CBF1   A6 83        11052   ob30     ldx  sa
     CBF3   BD 022B      11053            lda  lintab,x    ; set lindx table
     CBF6   09 40        11054            ora  #$40
     CBF8   9D 022B      11055            sta  lintab,x
                         11056
     CBFB   A4 82        11057            ldy  lindx
     CBFD   A9 FF        11058            lda  #$ff
     CBFF   99 0244      11059            sta  lstchr,y
                         11060
     CC02   A9 89        11061            lda  #rndrdy
     CC04   99 00F2      11062            sta  chnrdy,y    ; set channel ready
                         11063
     CC07   B9 00A7      11064            lda  buf0,y
     CC0A   99 023E      11065            sta  chndat,y    ; buffer # as 1st char
     CC0D   0A           11066            asl  a
     CC0E   AA           11067            tax
     CC0F   A9 01        11068            lda  #1
     CC11   95 99        11069            sta  buftab,x
     CC13   A9 0E        11070            lda  #dirtyp+dirtyp
     CC15   99 00EC      11071            sta  filtyp,y    ; set direct file type
     CC18   4C C194      11072            jmp  endcmd
```

```
Error Addr  Code           Seq    Source statement

                          11074
                          11075   ; block commands
                          11076
     CC1B   A0 00         11077   block   ldy  #0
     CC1D   A2 00         11078           ldx  #0
     CC1F   A9 2D         11079           lda  #'-        ; "-" separates cmd from subcmd
     CC21   20 C268       11080           jsr  parse      ; locate sub-cmd
     CC24   D0 0A         11081           bne  blk40
                          11082
     CC26   A9 31         11083   blk10   lda  #badcmd
     CC28   4C C1C8       11084           jmp  cmderr
                          11085
     CC2B   A9 30         11086   blk30   lda  #badsyn
     CC2D   4C C1C8       11087           jmp  cmderr
                          11088
     CC30   8A            11089   blk40   txa
     CC31   D0 F8         11090           bne  blk30
                          11091
     CC33   A2 05         11092           ldx  #nbcmds-1  ; find command
     CC35   B9 0200       11093           lda  cmdbuf,y
     CC38   DD CC5D       11094   blk50   cmp  bctab,x
     CC3B   F0 05         11095           beq  blk60
     CC3D   CA            11096           dex
     CC3E   10 F8         11097           bpl  blk50
     CC40   30 E4         11098           bmi  blk10
                          11099
     CC42                 11100   blk60
     CC42   8A            11101           txa
     CC43   09 80         11102           ora  #$80
     CC45   8D 022A       11103           sta  cmdnum
     CC48   20 CC6F       11104           jsr  blkpar     ; parse parms
                          11105
     CC4B   AD 022A       11106           lda  cmdnum
     CC4E   0A            11107           asl  a
     CC4F   AA            11108           tax
     CC50   BD CC64       11109           lda  bcjmp+1,x
     CC53   85 70         11110           sta  temp+1
     CC55   BD CC63       11111           lda  bcjmp,x
     CC58   85 6F         11112           sta  temp
                          11113
     CC5A   6C 006F       11114           jmp  (temp)     ; goto command
                          11115
     CC5D   41 46 52      11116   bctab    .byte  'AFRWEP'
     CC60   57 45 50
            =0006         11117   nbcmds   =*-bctab
                          11118
     CC63   CD03          11119   bcjmp    .word blkalc   ; block-allocate
     CC65   CCF5          11120            .word blkfre   ; block-free
     CC67   CD56          11121            .word blkrd    ; block-read
     CC69   CD73          11122            .word blkwt    ; block-write
```

```
Error Addr  Code           Seq    Source statement

     CC6B   CDA3         11123             .word blkexc    ; block-execute
     CC6D   CDBD         11124             .word blkptr    ; block-pointer
                         11125
     CC6F   A0 00        11126   blkpar    ldy  #0         ; parse block parms
     CC71   A2 00        11127             ldx  #0
     CC73   A9 3A        11128             lda  #':
     CC75   20 C268      11129             jsr  parse
     CC78   D0 02        11130             bne  bp05       ; found ":"
                         11131
     CC7A   A0 03        11132             ldy  #3         ; else char #3 is beginning
     CC7C   B9 0200      11133   bp05      lda  cmdbuf,y
     CC7F   C9 20        11134             cmp  #'
     CC81   F0 08        11135             beq  bp10
                         11136
     CC83   C9 1D        11137             cmp  #29        ; skip character
     CC85   F0 04        11138             beq  bp10
                         11139
     CC87   C9 2C        11140             cmp  #',
     CC89   D0 07        11141             bne  bp20
                         11142
     CC8B   C8           11143   bp10      iny
     CC8C   CC 0274      11144             cpy  cmdsiz
     CC8F   90 EB        11145             bcc  bp05
                         11146
     CC91   60           11147             rts             ; that's all
                         11148
     CC92   20 CCA1      11149   bp20      jsr  aschex
     CC95   EE 0277      11150             inc  f1cnt
     CC98   AC 0279      11151             ldy  f2ptr
     CC9B   E0 04        11152             cpx  #mxfils-1
     CC9D   90 EC        11153             bcc  bp10
                         11154
     CC9F   B0 8A        11155             bcs  blk30      ; bad syntax
                         11156
                         11157   ;  convert ascii to hex (binary)
                         11158   ;  & store conversion in tables
                         11159   ;  .y= ptr into cmdbuf
     CCA1   A9 00        11160   aschex    lda  #0
     CCA3   85 6F        11161             sta  temp
     CCA5   85 70        11162             sta  temp+1
     CCA7   85 72        11163             sta  temp+3
                         11164
     CCA9   A2 FF        11165             ldx  #$ff
     CCAB   B9 0200      11166   ah10      lda  cmdbuf,y   ; test for dec #
     CCAE   C9 40        11167             cmp  #$40
     CCB0   B0 18        11168             bcs  ah20       ; non-numeric terminates
     CCB2   C9 30        11169             cmp  #$30
     CCB4   90 14        11170             bcc  ah20       ; non-numeric
                         11171
     CCB6   29 0F        11172             and  #$f
```

```
Error Addr  Code           Seq    Source statement

     CCB8   48            11173            pha
     CCB9   A5 70         11174            lda  temp+1      ; shift digits (*10)
     CCBB   85 71         11175            sta  temp+2
     CCBD   A5 6F         11176            lda  temp
     CCBF   85 70         11177            sta  temp+1
     CCC1   68            11178            pla
     CCC2   85 6F         11179            sta  temp
     CCC4   C8            11180            iny
     CCC5   CC 0274       11181            cpy  cmdsiz
     CCC8   90 E1         11182            bcc  ah10        ; still in string
                          11183
     CCCA   8C 0279       11184   ah20     sty  f2ptr       ; convert digits to...
     CCCD   18            11185            clc              ; ...binary by dec table
     CCCE   A9 00         11186            lda  #0
                          11187
     CCD0   E8            11188   ah30     inx
     CCD1   E0 03         11189            cpx  #3
     CCD3   B0 0F         11190            bcs  ah40
                          11191
     CCD5   B4 6F         11192            ldy  temp,x
     CCD7   88            11193   ah35     dey
     CCD8   30 F6         11194            bmi  ah30
                          11195
     CCDA   7D CCF2       11196            adc  dectab,x
     CCDD   90 F8         11197            bcc  ah35
                          11198
     CCDF   18            11199            clc
     CCE0   E6 72         11200            inc  temp+3
     CCE2   D0 F3         11201            bne  ah35
                          11202
     CCE4   48            11203   ah40     pha
     CCE5   AE 0277       11204            ldx  flcnt
     CCE8   A5 72         11205            lda  temp+3
     CCEA   9D 0280       11206            sta  filtrk,x    ; store result in table
     CCED   68            11207            pla
     CCEE   9D 0285       11208            sta  filsec,x
     CCF1   60            11209            rts
                          11210
     CCF2   01 0A 64      11211   dectab    .byte 1,10,100 ; decimal table
                          11212
                          11213   ;block-free
     CCF5   20 CDF5       11214   blkfre   jsr  blktst
     CCF8   20 EF5F       11215            jsr  frets
     CCFB   4C C194       11216            jmp  endcmd
                          11217
                          11218   ;block-allocate
                          11219
     CCFE   A9 01         11220            lda  #1
     CD00   8D 02F9       11221            sta  wbam
     CD03                 11222   blkalc
```

```
Error Addr  Code           Seq    Source statement

     CD03   20 CDF5      11223            jsr  blktst
                         11224
     CD06                11225   ba10
     CD06   A5 81        11226            lda  sector
     CD08   48           11227            pha
     CD09   20 F1FA      11228            jsr  getsec
     CD0C   F0 0B        11229            beq  ba15        ; none greater on this track
     CD0E   68           11230            pla
     CD0F   C5 81        11231            cmp  sector
     CD11   D0 19        11232            bne  ba30        ; requested sector not avail
     CD13   20 EF90      11233            jsr  wused
     CD16   4C C194      11234            jmp  endcmd
                         11235
     CD19                11236   ba15
     CD19   68           11237            pla              ; pop stack
     CD1A                11238   ba20
     CD1A   A9 00        11239            lda  #0
     CD1C   85 81        11240            sta  sector
     CD1E   E6 80        11241            inc  track
     CD20   A5 80        11242            lda  track
     CD22   CD 02AC      11243            cmp  maxtrk
     CD25   B0 0A        11244            bcs  ba40        ; gone all the way
                         11245
     CD27   20 F1FA      11246            jsr  getsec
     CD2A   F0 EE        11247            beq  ba20
     CD2C                11248   ba30
     CD2C   A9 65        11249            lda  #noblk
     CD2E   20 E645      11250            jsr  cmder2
     CD31                11251   ba40
     CD31   A9 65        11252            lda  #noblk
     CD33   20 C1C8      11253            jsr  cmderr      ; t=0,s=0 :none left
                         11254
                         11255
                         11256   ; block read subs
     CD36   20 CDF2      11257   blkrd2   jsr  bkotst     ; test parms
     CD39   4C D460      11258            jmp  drtrd
                         11259
     CD3C   20 D12F      11260   getsim   jsr  getpre     ; get byte w/o inc
     CD3F   A1 99        11261            lda  (buftab,x)
     CD41   60           11262            rts
                         11263
                         11264   ; block read
     CD42   20 CD36      11265   blkrd3   jsr  blkrd2
     CD45   A9 00        11266            lda  #0
     CD47   20 D4C8      11267            jsr  setpnt
     CD4A   20 CD3C      11268            jsr  getsim      ; y=lindx
                         11269
                         11270
     CD4D   99 0244      11271            sta  lstchr,y
     CD50   A9 89        11272            lda  #rndrdy
```

```
Error Addr  Code           Seq    Source statement

     CD52   99 00F2      11273            sta  chnrdy,y
     CD55   60           11274            rts
     CD56                11275   blkrd
     CD56   20 CD42      11276            jsr  blkrd3
     CD59   20 D3EC      11277            jsr  rnget1
     CD5C   4C C194      11278            jmp  endcmd
                         11279
                         11280   ;user direct read, lstchr=$ff
     CD5F                11281   ublkrd
     CD5F   20 CC6F      11282            jsr  blkpar
     CD62   20 CD42      11283            jsr  blkrd3
     CD65   B9 0244      11284            lda  lstchr,y
     CD68   99 023E      11285            sta  chndat,y
     CD6B   A9 FF        11286            lda  #$ff
     CD6D   99 0244      11287            sta  lstchr,y
     CD70   4C C194      11288            jmp  endcmd     ; (rts)
                         11289
                         11290   ;block-write
     CD73   20 CDF2      11291   blkwt    jsr  bkotst
                         11292
     CD76   20 D4E8      11293            jsr  getpnt
     CD79   A8           11294            tay
     CD7A   88           11295            dey
     CD7B   C9 02        11296            cmp  #2
     CD7D   B0 02        11297            bcs  bw10
     CD7F   A0 01        11298            ldy  #1
                         11299
     CD81   A9 00        11300   bw10     lda  #0         ; set record size
     CD83   20 D4C8      11301            jsr  setpnt
     CD86   98           11302            tya
     CD87   20 CFF1      11303            jsr  putbyt
     CD8A   8A           11304            txa
     CD8B   48           11305            pha
                         11306
     CD8C   20 D464      11307   bw20     jsr  drtwrt     ; write block
     CD8F   68           11308            pla
     CD90   AA           11309            tax
                         11310
                         11311   ;        jsr  rnget2
     CD91   20 FFAE      11312            jsr  ptch15     ; fix for block read *rom ds 01/22/85*
                         11313
     CD94   4C C194      11314            jmp  endcmd
                         11315
                         11316   ;user dirct write, no lstchr
     CD97   20 CC6F      11317   ublkwt   jsr  blkpar
     CD9A   20 CDF2      11318            jsr  bkotst
     CD9D   20 D464      11319            jsr  drtwrt
     CDA0   4C C194      11320            jmp  endcmd
                         11321
                         11322   ;block-execute
```

```
Error Addr  Code           Seq    Source statement

     CDA3                 11323   blkexc
     CDA3   20 F258       11324           jsr  killp       ; kill protect
     CDA6   20 CD36       11325           jsr  blkrd2      ; read block & execute
     CDA9   A9 00         11326           lda  #0
                          11327
     CDAB   85 6F         11328   be05    sta  temp
     CDAD   A6 F9         11329           ldx  jobnum
     CDAF   BD FEE0       11330           lda  bufind,x
     CDB2   85 70         11331           sta  temp+1
     CDB4   20 CDBA       11332           jsr  be10        ; indirect jsr
     CDB7   4C C194       11333           jmp  endcmd
                          11334
     CDBA   6C 006F       11335   be10    jmp  (temp)
                          11336
                          11337   ;buffer-pointer, set buffer pointer
     CDBD   20 CDD2       11338   blkptr  jsr  buftst
     CDC0   A5 F9         11339           lda  jobnum
     CDC2   0A            11340           asl  a
     CDC3   AA            11341           tax
     CDC4   AD 0286       11342           lda  filsec+1
     CDC7   95 99         11343           sta  buftab,x
     CDC9   20 D12F       11344           jsr  getpre
     CDCC   20 D3EE       11345           jsr  rnget2      ; set up get
     CDCF   4C C194       11346           jmp  endcmd
                          11347
                          11348   ;test for allocated buffer..
                          11349   ;  ..related to sa
     CDD2   A6 D3         11350   buftst  ldx  fiptr
     CDD4   E6 D3         11351           inc  fiptr
     CDD6   BD 0285       11352           lda  filsec,x
     CDD9   A8            11353           tay
     CDDA   88            11354           dey
     CDDB   88            11355           dey
     CDDC   C0 0C         11356           cpy  #$c         ;  set limit to # of sas
     CDDE   90 05         11357           bcc  bt20
                          11358
     CDE0   A9 70         11359   bt15    lda  #nochnl
     CDE2   4C C1C8       11360           jmp  cmderr
                          11361
     CDE5   85 83         11362   bt20    sta  sa
     CDE7   20 D0EB       11363           jsr  fndrch
     CDEA   B0 F4         11364           bcs  bt15
     CDEC   20 DF93       11365           jsr  getact
     CDEF   85 F9         11366           sta  jobnum
     CDF1   60            11367           rts
                          11368
                          11369   ;test block operation parms
     CDF2   20 CDD2       11370   bkotst  jsr  buftst
                          11371   ;
                          11372   ;test for legal block &..
```

```
Error Addr  Code           Seq    Source statement

                          11373   ;  ..set up drv, trk, sec
     CDF5   A6 D3         11374   blktst  ldx  f1ptr
     CDF7   BD 0285       11375           lda  filsec,x
     CDFA   29 01         11376           and  #1
     CDFC   85 7F         11377           sta  drvnum
     CDFE   BD 0287       11378           lda  filsec+2,x
     CE01   85 81         11379           sta  sector
     CE03   BD 0286       11380           lda  filsec+1,x
     CE06   85 80         11381           sta  track
     CE08                 11382   bt05
     CE08   20 D55F       11383           jsr  tschk
     CE0B   4C C100       11384           jmp  setlds     ; (rts)
                          11385
                          11386   ; rsr 1/19/80 add autoi to #cmd
                          11387           .include  fndrel
```

```
Error Addr  Code           Seq   Source statement

                          11389           .subttl 'fndrel.src'
                          11390   ;*********************************
                          11391   ;*
                          11392   ;* find relative file
                          11393   ;*
                          11394   ;* version 2.5
                          11395   ;*
                          11396   ;*
                          11397   ;* inputs
                          11398   ;*  recl   - 1byte=lo record #
                          11399   ;*  rech   - 1byte=hi record #
                          11400   ;*  rs     - 1byte=record size
                          11401   ;*  recptr - 1byte=first byte
                          11402   ;*            wanted from record
                          11403   ;*
                          11404   ;* outputs
                          11405   ;*  ssnum  - 1byte=side sector #
                          11406   ;*  ssind  - 1byte=index into ss
                          11407   ;*  relptr - 1byte=ptr to first
                          11408   ;*            byte wanted
                          11409   ;*
                          11410   ;*********************************
                          11411
     CE0E   20 CE2C       11412   fndrel  jsr  mulply     ; result=rn*rs+rp
     CE11   20 CE6E       11413           jsr  div254     ; divide by 254
     CE14   A5 90         11414           lda  accum+1    ; save remainder
     CE16   85 D7         11415           sta  relptr
     CE18   20 CE71       11416           jsr  div120     ; divide by 120
     CE1B   E6 D7         11417           inc  relptr
     CE1D   E6 D7         11418           inc  relptr
     CE1F   A5 8B         11419           lda  result     ; save quotient
     CE21   85 D5         11420           sta  ssnum
     CE23   A5 90         11421           lda  accum+1    ; save remainder
     CE25   0A            11422           asl  a          ; calc index into ss
     CE26   18            11423           clc
     CE27   69 10         11424           adc  #16        ; skip link table
     CE29   85 D6         11425           sta  ssind
     CE2B   60            11426           rts
                          11427
                          11428
                          11429
                          11430   ; multiply
                          11431
                          11432   ; result=recnum*rs+recptr
                          11433
                          11434   ;  destroys a,x
                          11435
     CE2C   20 CED9       11436   mulply  jsr  zerres     ; result=0
     CE2F   85 92         11437           sta  accum+3    ; a=0
     CE31   A6 82         11438           ldx  lindx      ; get index
```

```
Error Addr  Code           Seq    Source statement

     CE33   B5 B5        11439            lda  recl,x
     CE35   85 90        11440            sta  accum+1
     CE37   B5 BB        11441            lda  rech,x
     CE39   85 91        11442            sta  accum+2
     CE3B   D0 04        11443            bne  mul25       ; adjust for rec #1 &...
     CE3D   A5 90        11444            lda  accum+1     ; ...#0 = 1st rec
     CE3F   F0 0B        11445            beq  mul50
     CE41   A5 90        11446   mul25    lda  accum+1
     CE43   38           11447            sec
     CE44   E9 01        11448            sbc  #1
     CE46   85 90        11449            sta  accum+1
     CE48   B0 02        11450            bcs  mul50
     CE4A   C6 91        11451            dec  accum+2
     CE4C                11452   mul50
     CE4C   B5 C7        11453            lda  rs,x        ; copy recsiz
     CE4E   85 6F        11454            sta  temp
     CE50   46 6F        11455   mul100   lsr  temp        ; do an add ?
     CE52   90 03        11456            bcc  mul200      ; no
     CE54   20 CEED      11457            jsr  addres      ; result=result+accum+1,2,3
     CE57   20 CEE5      11458   mul200   jsr  accx2       ; 2*(accum+1,2,3)
     CE5A   A5 6F        11459            lda  temp        ; done ?
     CE5C   D0 F2        11460            bne  mul100      ; no
     CE5E   A5 D4        11461            lda  recptr      ; add in last bit
     CE60   18           11462            clc
     CE61   65 8B        11463            adc  result
     CE63   85 8B        11464            sta  result
     CE65   90 06        11465            bcc  mul400      ; skip no carry
     CE67   E6 8C        11466            inc  result+1
     CE69   D0 02        11467            bne  mul400
     CE6B   E6 8D        11468            inc  result+2
     CE6D   60           11469   mul400   rts
                         11470
                         11471
                         11472
                         11473   ; divide
                         11474
                         11475   ; result=quotient ,accum+1=remainder
                         11476
                         11477   ;  destroys a,x
                         11478
     CE6E   A9 FE        11479   div254   lda  #254       ; divide by 254
     CE70   2C           11480            .byte  skip2    ; skip two bytes
     CE71   A9 78        11481   div120   lda  #120       ; divide by 120
     CE73   85 6F        11482            sta  temp       ; save divisor
     CE75   A2 03        11483            ldx  #3         ; swap accum+1,2,3 with
     CE77   B5 8F        11484   div100   lda  accum,x    ; result,1,2
     CE79   48           11485            pha
     CE7A   B5 8A        11486            lda  result-1,x
     CE7C   95 8F        11487            sta  accum,x
     CE7E   68           11488            pla
```

```
Error Addr  Code           Seq    Source statement

     CE7F   95 8A        11489            sta  result-1,x
     CE81   CA           11490            dex
     CE82   DO F3        11491            bne  div100
     CE84   20 CED9      11492            jsr  zerres      ; result=0
     CE87   A2 00        11493   div150   ldx  #0
     CE89   B5 90        11494   div200   lda  accum+1,x   ; divide by 256
     CE8B   95 8F        11495            sta  accum,x
     CE8D   E8           11496            inx
     CE8E   E0 04        11497            cpx  #4          ; done ?
     CE90   90 F7        11498            bcc  div200      ; no
     CE92   A9 00        11499            lda  #0          ; zero hi byte
     CE94   85 92        11500            sta  accum+3
     CE96   24 6F        11501            bit  temp        ; a div120 ?
     CE98   30 09        11502            bmi  div300      ; no
     CE9A   06 8F        11503            asl  accum       ; only divide by 128
     CE9C   08           11504            php              ; save carry
     CE9D   46 8F        11505            lsr  accum       ; normalize
     CE9F   28           11506            plp              ; restore carry
     CEA0   20 CEE6      11507            jsr  acc200      ; 2*(x/256)=x/128
     CEA3   20 CEED      11508   div300   jsr  addres      ; total a quotient
     CEA6   20 CEE5      11509            jsr  accx2       ; a=2*a
     CEA9   24 6F        11510            bit  temp        ; a div120 ?
     CEAB   30 03        11511            bmi  div400      ; no
     CEAD   20 CEE2      11512            jsr  accx4       ; a=4*(2*a)=8*a
     CEB0   A5 8F        11513   div400   lda  accum       ; add in remainder
     CEB2   18           11514            clc
     CEB3   65 90        11515            adc  accum+1
     CEB5   85 90        11516            sta  accum+1
     CEB7   90 06        11517            bcc  div500
     CEB9   E6 91        11518            inc  accum+2
     CEBB   DO 02        11519            bne  div500
     CEBD   E6 92        11520            inc  accum+3
     CEBF   A5 92        11521   div500   lda  accum+3     ; test < 256
     CEC1   05 91        11522            ora  accum+2
     CEC3   DO C2        11523            bne  div150      ; crunch some more
     CEC5   A5 90        11524            lda  accum+1     ; is remainder < divisor
     CEC7   38           11525            sec
     CEC8   E5 6F        11526            sbc  temp
     CECA   90 0C        11527            bcc  div700      ; yes
     CECC   E6 8B        11528            inc  result      ; no - fix result
     CECE   DO 06        11529            bne  div600
     CED0   E6 8C        11530            inc  result+1
     CED2   DO 02        11531            bne  div600
     CED4   E6 8D        11532            inc  result+2
     CED6   85 90        11533   div600   sta  accum+1     ; new remainder
     CED8   60           11534   div700   rts
                         11535
                         11536
                         11537
                         11538   ; zero result
```

```
Error Addr  Code           Seq    Source statement

                          11539
     CED9   A9 00         11540   zerres  lda  #0
     CEDB   85 8B         11541           sta  result
     CEDD   85 8C         11542           sta  result+1
     CEDF   85 8D         11543           sta  result+2
     CEE1   60            11544           rts
                          11545
                          11546
                          11547   ; multiply accum by 4
                          11548
     CEE2   20 CEE5       11549   accx4   jsr  accx2
                          11550
                          11551   ; multiply accum by 2
                          11552
     CEE5   18            11553   accx2   clc
     CEE6   26 90         11554   acc200  rol  accum+1
     CEE8   26 91         11555           rol  accum+2
     CEEA   26 92         11556           rol  accum+3
     CEEC   60            11557           rts
                          11558
                          11559
                          11560
                          11561   ; add accum to result
                          11562
                          11563   ;  result=result+accum+1,2,3
                          11564
     CEED   18            11565   addres  clc
     CEEE   A2 FD         11566           ldx  #$fd
     CEF0   B5 8E         11567   add100  lda  result+3,x
     CEF2   75 93         11568           adc  accum+4,x
     CEF4   95 8E         11569           sta  result+3,x
     CEF6   E8            11570           inx
     CEF7   D0 F7         11571           bne  add100
     CEF9   60            11572           rts
                          11573           .include  tst2
```

```
Error Addr  Code           Seq    Source statement

                          11575           .subttl 'tst2.src' .
                          11576
                          11577   ;*******************************
                          11578   ;************  l r u i n t  ****
                          11579   ;*******************************
                          11580   ;
                          11581   ; initialize the lru tabel
                          11582   ;
                          11583   ;*******************************
                          11584
     CEFA                 11585   lruint
     CEFA   A2 00         11586           ldx  #0
     CEFC                 11587   lruilp
     CEFC   8A            11588           txa
     CEFD   95 FA         11589           sta  lrutbl,x
     CEFF   E8            11590           inx
     CF00   E0 04         11591           cpx  #cmdchn
     CF02   D0 F8         11592           bne  lruilp
                          11593
     CF04   A9 06         11594           lda  #blindx
     CF06   95 FA         11595           sta  lrutbl,x
     CF08   60            11596           rts
                          11597
                          11598   ;*******************************
                          11599   ;************  l r u u p d  ***
                          11600   ;*******************************
                          11601
                          11602   ; least recently used table update
                          11603
                          11604   ;     input parameters:
                          11605   ;         lindx - current channel
                          11606
                          11607   ;     output parameters:
                          11608   ;         lrutbl - updated
                          11609
                          11610   ;*******************************
                          11611
     CF09   A0 04         11612   lruupd  ldy  #cmdchn
     CF0B   A6 82         11613           ldx  lindx
     CF0D   B9 00FA       11614   lrulp1  lda  lrutbl,y
     CF10   96 FA         11615           stx  lrutbl,y
     CF12   C5 82         11616           cmp  lindx
     CF14   F0 07         11617           beq  lruext
     CF16   88            11618           dey
     CF17   30 E1         11619           bmi  lruint
     CF19   AA            11620           tax
     CF1A   4C CF0D       11621           jmp  lrulp1
     CF1D   60            11622   lruext  rts
                          11623
                          11624   ;*******************************
```

```
Error Addr  Code           Seq    Source statement

                          11625   ;*************  d b l b u f  ***
                          11626   ;*******************************
                          11627
                          11628   ; double buffer
                          11629   ; rtn to switch the active and
                          11630   ; inactive buffers
                          11631
                          11632   ;*******************************
                          11633
     CF1E                 11634   dblbuf
     CF1E   20 CF09       11635           jsr  lruupd
     CF21   20 DFB7       11636           jsr  getina
     CF24   D0 46         11637           bne  dbl15
     CF26   20 D1D3       11638           jsr  setdrn
     CF29   20 D28E       11639           jsr  getbuf
     CF2C   30 48         11640           bmi  dbl30      ; no buffers
     CF2E   20 DFC2       11641           jsr  putina     ; store inactive buff #
     CF31   A5 80         11642           lda  track
     CF33   48            11643           pha
     CF34   A5 81         11644           lda  sector
     CF36   48            11645           pha
     CF37   A9 01         11646           lda  #1
     CF39   20 D4F6       11647           jsr  drdbyt
     CF3C   85 81         11648           sta  sector
     CF3E   A9 00         11649           lda  #0
     CF40   20 D4F6       11650           jsr  drdbyt
     CF43   85 80         11651           sta  track
                          11652
     CF45   F0 1F         11653           beq  dbl10
                          11654
     CF47   20 D125       11655           jsr  typfil
     CF4A   F0 0B         11656           beq  dbl05      ; it's rel
                          11657
     CF4C   20 DDAB       11658           jsr  tstwrt
     CF4F   D0 06         11659           bne  dbl05      ; read ahead
                          11660
     CF51   20 CF8C       11661           jsr  tglbuf     ; just switch on write
     CF54   4C CF5D       11662           jmp  dbl08
                          11663
     CF57                 11664   dbl05
     CF57   20 CF8C       11665           jsr  tglbuf
     CF5A   20 DE57       11666           jsr  rdab
     CF5D                 11667   dbl08
     CF5D   68            11668           pla
     CF5E   85 81         11669           sta  sector
     CF60   68            11670           pla
     CF61   85 80         11671           sta  track
     CF63   4C CF6F       11672           jmp  dbl20
     CF66                 11673   dbl10
     CF66   68            11674           pla
```

```
Error Addr  Code           Seq    Source statement

     CF67   85 81        11675            sta  sector
     CF69   68           11676            pla
     CF6A   85 80        11677            sta  track
     CF6C   20 CF8C      11678   dbl15    jsr  tglbuf
     CF6F   20 DF93      11679   dbl20    jsr  getact
     CF72   AA           11680            tax
     CF73   4C D599      11681            jmp  watjob
                         11682
                         11683   ; there are no buffers to steal
                         11684
     CF76                11685   dbl30
     CF76   A9 70        11686            lda  #nochnl
     CF78   4C C1C8      11687            jmp  cmderr
                         11688
                         11689   ;*********************************
                         11690
     CF7B                11691   dbset
     CF7B   20 CF09      11692            jsr  lruupd
     CF7E   20 DFB7      11693            jsr  getina
     CF81   D0 08        11694            bne  dbs10
     CF83   20 D28E      11695            jsr  getbuf
     CF86   30 EE        11696            bmi  dbl30       ; no buffers
     CF88   20 DFC2      11697            jsr  putina      ; store inactive buff #
     CF8B                11698   dbs10
     CF8B   60           11699            rts
                         11700   ;*********************************
                         11701   ;************  t g l b u f  *****
                         11702   ;*********************************
                         11703
                         11704   ; toggle the inactive and active
                         11705   ; buffers.
                         11706
                         11707   ;     input parameters:
                         11708   ;          lindx  -  channel #
                         11709
                         11710   ;*********************************
                         11711
     CF8C   A6 82        11712   tglbuf   ldx  lindx
     CF8E   B5 A7        11713            lda  buf0,x
     CF90   49 80        11714            eor  #$80
     CF92   95 A7        11715            sta  buf0,x
     CF94   B5 AE        11716            lda  buf1,x
     CF96   49 80        11717            eor  #$80
     CF98   95 AE        11718            sta  buf1,x
     CF9A   60           11719            rts
                         11720
                         11721
     CF9B                11722   pibyte
     CF9B   A2 12        11723            ldx  #iwsa
     CF9D   86 83        11724            stx  sa
```

```
Error Addr  Code          Seq    Source statement

     CF9F   20 D107      11725            jsr  fndwch
     CFA2   20 C100      11726            jsr  setlds
     CFA5   20 D125      11727            jsr  typfil
     CFA8   90 05        11728            bcc  pbyte
     CFAA   A9 20        11729            lda  #ovrflo
     CFAC   20 DD9D      11730            jsr  clrflg
     CFAF                11731  pbyte
     CFAF   A5 83        11732            lda  sa
     CFB1   C9 0F        11733            cmp  #15
     CFB3   F0 23        11734            beq  142
     CFB5   D0 08        11735            bne  140
                         11736
                         11737  ; main routine to write to chanl
                         11738
     CFB7   A5 84        11739  put       lda  orgsa       ; is chanl cmd or data
     CFB9   29 8F        11740            and  #$8f
     CFBB   C9 0F        11741            cmp  #15         ; <15
     CFBD   B0 19        11742            bcs  142
     CFBF   20 D125      11743  140       jsr  typfil      ; data byte to store
     CFC2   B0 05        11744            bcs  141         ; branch if rnd
     CFC4   A5 85        11745            lda  data        ; seq file
     CFC6   4C D19D      11746            jmp  wrtbyt      ; write byte to chanl
     CFC9                11747  141
     CFC9   D0 03        11748            bne  146
     CFCB   4C E0AB      11749            jmp  wrtrel
     CFCE   A5 85        11750  146       lda  data        ; rnd file write
     CFD0   20 CFF1      11751            jsr  putbyt      ; write to chanl
     CFD3   A4 82        11752            ldy  lindx       ;  prepare nxt byte
     CFD5   4C D3EE      11753            jmp  rnget2
     CFD8   A9 04        11754  142       lda  #cmdchn     ; write to cmd chanl
     CFDA   85 82        11755            sta  lindx
     CFDC   20 D4E8      11756            jsr  getpnt      ; test if comm and buffer full
     CFDF   C9 2A        11757            cmp  #<cmdbuf+cmdlen+1
     CFE1   F0 05        11758            beq  150         ; it is full (>cmdlen)
     CFE3   A5 85        11759            lda  data        ; not full yet
     CFE5   20 CFF1      11760            jsr  putbyt      ; store the byte
     CFE8   A5 F8        11761  150       lda  eoiflg      ; tst if lst byte of msg
     CFEA   F0 01        11762            beq  145         ; it is
     CFEC   60           11763            rts              ; not yet , return
     CFED                11764  145
     CFED   EE 0255      11765            inc  cmdwat      ; set cmd waiting flag
     CFF0   60           11766            rts
                         11767
                         11768  ; put .a into active buffer of lindx
                         11769
     CFF1   48           11770  putbyt    pha              ;  save .a
     CFF2   20 DF93      11771            jsr  getact      ; get active buf#
     CFF5   10 06        11772            bpl  putb1       ; brach if there is one
     CFF7   68           11773            pla              ; no buffer error
     CFF8   A9 61        11774            lda  #filnop
```

```
Error Addr  Code          Seq    Source statement

     CFFA   4C C1C8      11775            jmp  cmderr      ;  jmp to error routine
     CFFD   0A           11776   putb1    asl  a           ; save the byte in buffer
     CFFE   AA           11777            tax
     CFFF   68           11778            pla
     D000   81 99        11779            sta  (buftab,x)
     D002   F6 99        11780            inc  buftab,x    ;  inc the buffer pointer
     D004   60           11781            rts              ; z=1 if last char slot in buffer
                         11782
                         11783   ; find the active buffer # (lindx)
                         11784
                         11785
                         11786   ;initialize drives (command)
     D005   20 C1D1      11787   intdrv   jsr  simprs
     D008   20 D042      11788            jsr  initdr
     D00B   4C C194      11789   id20     jmp  endcmd
                         11790
                         11791   ; initialize drive (drvnum)
                         11792
     D00E                11793   itrial
     D00E   20 F10F      11794            jsr  bam2a
     D011   A8           11795            tay              ; bam lindx from buf0
     D012   B6 A7        11796            ldx  buf0,y
     D014   E0 FF        11797            cpx  #$ff
     D016   D0 14        11798            bne  it30        ; valid buffer #
                         11799
     D018   48           11800            pha              ; save bam-lindx
     D019   20 D28E      11801            jsr  getbuf      ; get a buffer
     D01C   AA           11802            tax              ; test it
     D01D   10 05        11803            bpl  it20        ; no error
                         11804
     D01F   A9 70        11805            lda  #nochnl
     D021   20 E648      11806            jsr  cmder3
     D024                11807   it20
     D024   68           11808            pla
     D025   A8           11809            tay              ; restore bam-lindx
     D026   8A           11810            txa
     D027   09 80        11811            ora  #$80        ; set inactive
     D029   99 00A7      11812            sta  buf0,y
     D02C                11813   it30
     D02C   8A           11814            txa
     D02D   29 0F        11815            and  #$f         ; stip to buf #
     D02F   85 F9        11816            sta  jobnum
     D031   A2 00        11817            ldx  #0
     D033   86 81        11818            stx  sector
     D035   AE FE85      11819            ldx  dirtrk
     D038   86 80        11820            stx  track
     D03A   20 D6D3      11821            jsr  seth        ; set the bam header
     D03D   A9 B0        11822            lda  #seek
                         11823
                         11824   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          11825
     D03F   4C A6E5       11826            jmp  ptch24      ; *** rom ds 02/21/85 ***
                          11827    ;       jmp  dojob       ; do a seek
                          11828
                          11829    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          11875   ;
     D075                 11876   nfcalc
     D075   20 EF3A       11877           jsr  setbpt
                          11878
     D078   A0 04         11879           ldy  #4
     D07A   A9 00         11880           lda  #0
     D07C   AA            11881           tax             ; 0 hi byte
     D07D                 11882   numf1
     D07D   18            11883           clc
     D07E   71 6D         11884           adc  (bmpnt),y
     D080   90 01         11885           bcc  numf2
     D082   E8            11886           inx
     D083                 11887   numf2
     D083   C8            11888           iny
     D084   C8            11889           iny
     D085   C8            11890           iny
     D086   C8            11891           iny
     D087   C0 48         11892           cpy  #$48       ;  dont count the dir
     D089   F0 F8         11893           beq  numf2
     D08B   C0 90         11894           cpy  #$90
     D08D   D0 EE         11895           bne  numf1
     D08F   48            11896           pha
     D090   8A            11897           txa
     D091   A6 7F         11898           ldx  drvnum
     D093   9D 02FC       11899           sta  ndbh,x
     D096   68            11900           pla
                          11901
                          11902   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          11903
     D097   4C A951       11904           jmp  ptch40     ; *** rom ds 02/26/85 ****
                          11905   ;       sta  ndbl,x
     D09A   60            11906           rts
                          11907
                          11908   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          11909
                          11910   ; start double buffering
                          11911   ; use track,sector as starting block
                          11912
     D09B                 11913   strrd
     D09B   20 D6D0       11914           jsr  sethdr
     D09E   20 D0C3       11915           jsr  rdbuf
     D0A1   20 D599       11916           jsr  watjob
     D0A4   20 D137       11917           jsr  getbyt
     D0A7   85 80         11918           sta  track
     D0A9   20 D137       11919           jsr  getbyt
     D0AC   85 81         11920           sta  sector
     D0AE   60            11921           rts
                          11922
     D0AF                 11923   strdbl
     D0AF   20 D09B       11924           jsr  strrd
```

```
Error Addr  Code          Seq    Source statement

     D0B2   A5 80        11925            lda  track
     D0B4   D0 01        11926            bne  str1
     D0B6   60           11927            rts
     D0B7                11928    str1
     D0B7   20 CF1E      11929            jsr  dblbuf
     D0BA   20 D6D0      11930            jsr  sethdr
     D0BD   20 D0C3      11931            jsr  rdbuf
     D0C0   4C CF1E      11932            jmp  dblbuf
                         11933
                         11934    ; start a read job on track,sector
                         11935
     D0C3   A9 80        11936    rdbuf   lda  #read
     D0C5   D0 02        11937            bne  strtit
                         11938
                         11939    ; start a write job on track,sector
                         11940
     D0C7   A9 90        11941    wrtbuf  lda  #write
     D0C9   8D 024D      11942    strtit  sta  cmd
     D0CC   20 DF93      11943            jsr  getact
     D0CF   AA           11944            tax
     D0D0   20 D506      11945            jsr  setljb
     D0D3   8A           11946            txa
     D0D4   48           11947            pha
     D0D5   0A           11948            asl  a
     D0D6   AA           11949            tax
     D0D7   A9 00        11950            lda  #0
     D0D9   95 99        11951            sta  buftab,x
     D0DB   20 D125      11952            jsr  typfil
     D0DE   C9 04        11953            cmp  #4
     D0E0   B0 06        11954            bcs  wrtc1      ; not sequential type
                         11955
     D0E2   F6 B5        11956            inc  nbkl,x
     D0E4   D0 02        11957            bne  wrtc1
     D0E6   F6 BB        11958            inc  nbkh,x
     D0E8   68           11959    wrtc1   pla
     D0E9   AA           11960            tax
     D0EA   60           11961            rts
                         11962
                         11963
                         11964    ;*****************************
                         11965    ;*
                         11966    ;*   fndrch
                         11967    ;*****************************
                         11968    ;*
                         11969    ;*
     D0EB   A5 83        11970    fndrch  lda  sa
     D0ED   C9 13        11971            cmp  #maxsa+1
     D0EF   90 02        11972            bcc  fndc20
     D0F1   29 0F        11973            and  #$f
     D0F3                11974    fndc20
```

```
Error Addr  Code           Seq    Source statement

     DOF3   C9 OF        11975            cmp  #cmdsa
     DOF5   DO O2        11976            bne  fndc25
     DOF7   A9 10        11977            lda  #errsa
     DOF9                11978   fndc25
     DOF9   AA           11979            tax
     DOFA   38           11980            sec
     DOFB   BD O22B      11981            lda  lintab,x
     DOFE   30 O6        11982            bmi  fndc30
     D100   29 OF        11983            and  #$f
     D102   85 82        11984            sta  lindx
     D104   AA           11985            tax
     D105   18           11986            clc
     D106   60           11987   fndc30   rts
                         11988   ;*
                         11989   ;*
                         11990   ;*****************************
                         11991   ;*
                         11992   ;*    fndwch
                         11993   ;*
                         11994   ;*****************************
                         11995   ;*
                         11996   ;*
     D107   A5 83        11997   fndwch   lda  sa
     D109   C9 13        11998            cmp  #maxsa+1
     D10B   90 O2        11999            bcc  fndw13
     D10D   29 OF        12000            and  #$f
     D10F   AA           12001   fndw13   tax
     D110   BD O22B      12002            lda  lintab,x
     D113   A8           12003            tay
     D114   OA           12004            asl  a
     D115   90 OA        12005            bcc  fndw15
     D117   30 OA        12006            bmi  fndw20
     D119   98           12007   fndw10   tya
     D11A   29 OF        12008            and  #$0f
     D11C   85 82        12009            sta  lindx
     D11E   AA           12010            tax
     D11F   18           12011            clc
     D120   60           12012            rts
     D121   30 F6        12013   fndw15   bmi  fndw10
     D123   38           12014   fndw20   sec
     D124   60           12015            rts
                         12016
     D125                12017   typfil                    ; get file type
     D125   A6 82        12018            ldx  lindx
     D127   B5 EC        12019            lda  filtyp,x
     D129   4A           12020            lsr  a
     D12A   29 O7        12021            and  #7
     D12C   C9 O4        12022            cmp  #reltyp
     D12E   60           12023            rts
                         12024
```

```
Error Addr  Code           Seq    Source statement

     D12F   20 DF93      12025   getpre  jsr  getact
     D132   0A          12026           asl  a
     D133   AA          12027           tax
     D134   A4 82       12028           ldy  lindx
     D136   60          12029           rts
                        12030
                        12031   ; read byte from active buffer
                        12032   ; and set flag if last data byte
                        12033   ; if last then z=1 else z=0 ;
     D137   20 D12F      12034   getbyt  jsr  getpre
     D13A   B9 0244      12035           lda  lstchr,y
     D13D   F0 12        12036           beq  getb1
     D13F   A1 99        12037           lda  (buftab,x)
     D141   48          12038           pha
     D142   B5 99        12039           lda  buftab,x
     D144   D9 0244      12040           cmp  lstchr,y
     D147   D0 04        12041           bne  getb2
     D149   A9 FF        12042           lda  #$ff
     D14B   95 99        12043           sta  buftab,x
     D14D   68          12044   getb2   pla
     D14E   F6 99        12045           inc  buftab,x
     D150   60          12046           rts
     D151   A1 99        12047   getb1   lda  (buftab,x)
     D153   F6 99        12048           inc  buftab,x
     D155   60          12049           rts
                        12050
                        12051
                        12052   ; read a char from file and read next
                        12053   ; block of file if needed.
                        12054   ; set chnrdy=eoi if end of file
                        12055
     D156   20 D137      12056   rdbyt   jsr  getbyt
     D159   D0 36        12057           bne  rd3
     D15B   85 85        12058           sta  data
                        12059
     D15D   B9 0244      12060   rd0     lda  lstchr,y
     D160   F0 08        12061           beq  rd1
     D162   A9 80        12062           lda  #eoiout
     D164   99 00F2      12063   rd01    sta  chnrdy,y
     D167   A5 85        12064           lda  data
     D169   60          12065           rts
     D16A   20 CF1E      12066   rd1     jsr  dblbuf
     D16D   A9 00        12067           lda  #0
     D16F   20 D4C8      12068           jsr  setpnt
     D172   20 D137      12069           jsr  getbyt
     D175   C9 00        12070           cmp  #0
     D177   F0 19        12071           beq  rd4
     D179   85 80        12072           sta  track
     D17B   20 D137      12073           jsr  getbyt
     D17E   85 81        12074           sta  sector
```

```
Error Addr  Code          Seq    Source statement

     D180   20 CF1E      12075            jsr  dblbuf
     D183   20 D1D3      12076            jsr  setdrn
     D186   20 D6D0      12077            jsr  sethdr
     D189   20 D0C3      12078            jsr  rdbuf
     D18C   20 CF1E      12079            jsr  dblbuf
     D18F   A5 85        12080            lda  data
     D191   60           12081   rd3      rts
     D192   20 D137      12082   rd4      jsr  getbyt
     D195   A4 82        12083            ldy  lindx
     D197   99 0244      12084            sta  lstchr,y
     D19A   A5 85        12085            lda  data
     D19C   60           12086            rts
                         12087
                         12088
                         12089   ; write a char to chanl and write
                         12090   ; buffer out to disk if its full
                         12091
     D19D   20 CFF1      12092   wrtbyt   jsr  putbyt
     D1A0   F0 01        12093            beq  wrt0
     D1A2   60           12094            rts
                         12095
     D1A3   20 D1D3      12096   wrt0     jsr  setdrn
     D1A6   20 F11E      12097            jsr  nxtts
     D1A9   A9 00        12098            lda  #0
     D1AB   20 D4C8      12099            jsr  setpnt
     D1AE   A5 80        12100            lda  track
     D1B0   20 CFF1      12101            jsr  putbyt
     D1B3   A5 81        12102            lda  sector
     D1B5   20 CFF1      12103            jsr  putbyt
     D1B8   20 D0C7      12104            jsr  wrtbuf
     D1BB   20 CF1E      12105            jsr  dblbuf
     D1BE   20 D6D0      12106            jsr  sethdr
     D1C1   A9 02        12107            lda  #2
     D1C3   4C D4C8      12108            jmp  setpnt
                         12109
                         12110   ; inc pointer of active buffer
                         12111   ; by .a
                         12112
                         12113
     D1C6                12114   incptr                    ;  scott patch
     D1C6   85 6F        12115   incpnt   sta  temp
     D1C8   20 D4E8      12116            jsr  getpnt
     D1CB   18           12117            clc
     D1CC   65 6F        12118            adc  temp
     D1CE   95 99        12119            sta  buftab,x
     D1D0   85 94        12120            sta  dirbuf
     D1D2   60           12121            rts
                         12122
                         12123   ; set drvnum to drive indicated by
                         12124   ; lstjob of active buffer
```

```
Error Addr  Code           Seq    Source statement

                          12125
     D1D3   20 DF93       12126   setdrn  jsr  getact
     D1D6   AA            12127           tax
     D1D7   BD 025B       12128           lda  lstjob,x
     D1DA   29 01         12129           and  #1
     D1DC   85 7F         12130           sta  drvnum
     D1DE   60            12131           rts
                          12132           .include  tst3
```

```
Error Addr  Code           Seq    Source statement

                          12134            .subttl 'tst3.src'
                          12135
                          12136
                          12137    ; .a=#buffers needed
                          12138    ; sets up buffer # and allocates lindx
                          12139
     D1DF                 12140    getwch
     D1DF   38            12141            sec             ; set .c=1 indicate write
     D1E0   B0 01         12142            bcs  getr2
     D1E2                 12143    getrch
     D1E2   18            12144            clc             ; set .c=0 indicate read
     D1E3                 12145    getr2
     D1E3   08            12146            php             ; save r/w flag (.c)
     D1E4   85 6F         12147            sta  temp       ; save #bufs needed
     D1E6   20 D227       12148            jsr  frechn     ; free any channels
     D1E9   20 D37F       12149            jsr  fndlnx     ;  get next lindx open
     D1EC   85 82         12150            sta  lindx
     D1EE   A6 83         12151            ldx  sa
     D1F0   28            12152            plp
     D1F1   90 02         12153            bcc  getr55
     D1F3   09 80         12154    getr52  ora  #$80
     D1F5   9D 022B       12155    getr55  sta  lintab,x   ; save lindx in lintab
     D1F8                 12156    getbf
     D1F8   29 3F         12157            and  #$3f
     D1FA   A8            12158            tay             ; now get the buffers
     D1FB   A9 FF         12159            lda  #$ff
     D1FD   99 00A7       12160            sta  buf0,y
     D200   99 00AE       12161            sta  buf1,y
     D203   99 00CD       12162            sta  ss,y
     D206   C6 6F         12163    getr3   dec  temp
     D208   30 1C         12164            bmi  getr4
     D20A   20 D28E       12165            jsr  getbuf
     D20D   10 08         12166            bpl  getr5
     D20F   20 D25A       12167    gberr   jsr  relbuf     ;  error ,rel bufs
     D212   A9 70         12168            lda  #nochnl
     D214   4C C1C8       12169            jmp  cmderr
     D217   99 00A7       12170    getr5   sta  buf0,y
     D21A   C6 6F         12171            dec  temp
     D21C   30 08         12172            bmi  getr4
     D21E   20 D28E       12173            jsr  getbuf
     D221   30 EC         12174            bmi  gberr
     D223   99 00AE       12175            sta  buf1,y
     D226   60            12176    getr4   rts
                          12177
                          12178    ; free chanl associated with sa
                          12179    ; free read and write chanls
                          12180    ; dont free chanl 15
     D227   A5 83         12181    frechn  lda  sa
     D229   C9 0F         12182            cmp  #$f
     D22B   D0 01         12183            bne  freco
```

```
Error Addr  Code           Seq    Source statement

     D22D   60           12184            rts
                         12185
     D22E                12186   freco
     D22E                12187   frerd
     D22E   A6 83        12188   frewrt   ldx  sa
     D230   BD 022B      12189            lda  lintab,x
     D233   C9 FF        12190            cmp  #$ff
     D235   F0 22        12191            beq  fre25
     D237   29 3F        12192            and  #$3f
     D239   85 82        12193            sta  lindx
     D23B   A9 FF        12194            lda  #$ff
     D23D   9D 022B      12195            sta  lintab,x
     D240   A6 82        12196            ldx  lindx
     D242   A9 00        12197            lda  #0
     D244   95 F2        12198            sta  chnrdy,x
     D246   20 D25A      12199            jsr  relbuf
                         12200
                         12201   ; release the lindx
                         12202
     D249   A6 82        12203   relinx   ldx  lindx
     D24B   A9 01        12204            lda  #1
     D24D   CA           12205   rel15    dex
     D24E   30 03        12206            bmi  rel10
     D250   0A           12207            asl  a
     D251   D0 FA        12208            bne  rel15
     D253   0D 0256      12209   rel10    ora  linuse      ; 1=free 0=used
     D256   8D 0256      12210            sta  linuse
     D259   60           12211   fre25    rts
                         12212
                         12213   ; given sa , free its read chanl
                         12214
                         12215
                         12216   ; release buffers (lindx)
                         12217
     D25A   A6 82        12218   relbuf   ldx  lindx
     D25C   B5 A7        12219            lda  buf0,x
     D25E   C9 FF        12220            cmp  #$ff
     D260   F0 09        12221            beq  rel1
     D262   48           12222            pha
     D263   A9 FF        12223            lda  #$ff
     D265   95 A7        12224            sta  buf0,x
     D267   68           12225            pla
     D268   20 D2F3      12226            jsr  frebuf
     D26B   A6 82        12227   rel1     ldx  lindx
     D26D   B5 AE        12228            lda  buf1,x
     D26F   C9 FF        12229            cmp  #$ff
     D271   F0 09        12230            beq  rel2
     D273   48           12231            pha
     D274   A9 FF        12232            lda  #$ff
     D276   95 AE        12233            sta  buf1,x
```

```
Error Addr  Code           Seq    Source statement

     D278   68            12234            pla
     D279   20 D2F3       12235            jsr  frebuf
     D27C   A6 82         12236   rel2     ldx  lindx
     D27E   B5 CD         12237            lda  ss,x
     D280   C9 FF         12238            cmp  #$ff
     D282   F0 09         12239            beq  rel3
     D284   48            12240            pha
     D285   A9 FF         12241            lda  #$ff
     D287   95 CD         12242            sta  ss,x
     D289   68            12243            pla
     D28A   20 D2F3       12244            jsr  frebuf
     D28D   60            12245   rel3     rts
                          12246
                          12247   ; get a free buffer #
                          12248
                          12249   ;  regs destroyed: .a  .x
                          12250   ;  out:  .a,.x= buf # or $ff  if failed
                          12251   ;        .n= 1 if failed
                          12252
                          12253   ;     if successful init jobs & lstjob
                          12254
     D28E                 12255   getbuf
     D28E   98            12256            tya              ; save .y
     D28F   48            12257            pha
     D290   A0 01         12258            ldy  #1
     D292   20 D2BA       12259            jsr  fndbuf
     D295   10 0C         12260            bpl  gbf1        ; found one
                          12261
     D297   88            12262            dey
     D298   20 D2BA       12263            jsr  fndbuf
     D29B   10 06         12264            bpl  gbf1        ; found one
                          12265
     D29D   20 D339       12266            jsr  stlbuf      ; steal one
     D2A0   AA            12267            tax              ; test it
     D2A1   30 13         12268            bmi  gbf2        ; didn't find one
     D2A3                 12269   gbf1
     D2A3   B5 00         12270            lda  jobs,x
     D2A5   30 FC         12271            bmi  gbf1        ; wait for job free
     D2A7   A5 7F         12272            lda  drvnum
     D2A9   95 00         12273            sta  jobs,x      ; clear job queue
     D2AB   9D 025B       12274            sta  lstjob,x
     D2AE   8A            12275            txa
     D2AF   0A            12276            asl  a
     D2B0   A8            12277            tay
     D2B1   A9 02         12278            lda  #2
     D2B3   99 0099       12279            sta  buftab,y
     D2B6                 12280   gbf2
     D2B6   68            12281            pla
     D2B7   A8            12282            tay              ; restore .y
     D2B8   8A            12283            txa              ; exit with buf # in .a & cc set
```

```
Error Addr  Code           Seq    Source statement

     D2B9   60           12284            rts
                         12285
                         12286
                         12287    ;  find a free buf # & set bit in bufuse
                         12288    ;  all regs used
                         12289    ;  in:  .y= index into bufuse
                         12290    ;  out: .x= buf # or $ff  if failed
                         12291    ;       .n= 1 if failed
                         12292
     D2BA                12293    fndbuf
     D2BA   A2 07        12294            ldx  #7
     D2BC                12295    fb1
     D2BC   B9 024F      12296            lda  bufuse,y    ; search bufuse
     D2BF   3D EFE9      12297            and  bmask,x
     D2C2   F0 04        12298            beq  fb2         ; found a free one
                         12299
     D2C4   CA           12300            dex
     D2C5   10 F5        12301            bpl  fb1         ; until all bits are tested
                         12302
     D2C7   60           12303            rts
     D2C8                12304    fb2                      ; set it used
     D2C8   B9 024F      12305            lda  bufuse,y
     D2CB   5D EFE9      12306            eor  bmask,x     ; set bit
     D2CE   99 024F      12307            sta  bufuse,y
     D2D1   8A           12308            txa
     D2D2   88           12309            dey
     D2D3   30 03        12310            bmi  fb3         ; if .y then...
     D2D5   18           12311            clc              ; ...add 8 more
     D2D6   69 08        12312            adc  #8
     D2D8                12313    fb3
     D2D8   AA           12314            tax              ; leave in .x
     D2D9                12315    fri20
     D2D9   60           12316            rts
                         12317
     D2DA                12318    freiac                   ; free inactive buffer
     D2DA   A6 82        12319            ldx  lindx
     D2DC   B5 A7        12320            lda  buf0,x
     D2DE   30 09        12321            bmi  fri10
                         12322
     D2E0   8A           12323            txa
     D2E1   18           12324            clc
     D2E2   69 07        12325            adc  #mxchns+1
     D2E4   AA           12326            tax
     D2E5   B5 A7        12327            lda  buf0,x
     D2E7   10 F0        12328            bpl  fri20
     D2E9                12329    fri10
     D2E9   C9 FF        12330            cmp  #$ff
     D2EB   F0 EC        12331            beq  fri20
                         12332
     D2ED   48           12333            pha
```

```
Error Addr  Code           Seq    Source statement

     D2EE   A9 FF        12334            lda  #$ff
     D2F0   95 A7        12335            sta  buf0,x
     D2F2   68           12336            pla
                         12337
     D2F3   29 0F        12338    frebuf  and  #$f
     D2F5   A8           12339            tay
     D2F6   C8           12340            iny
     D2F7   A2 10        12341            ldx  #16
     D2F9   6E 0250      12342    freb1   ror  bufuse+1
     D2FC   6E 024F      12343            ror  bufuse
     D2FF   88           12344            dey
     D300   D0 01        12345            bne  freb2
     D302   18           12346            clc
     D303   CA           12347    freb2   dex
     D304   10 F3        12348            bpl  freb1
     D306   60           12349            rts
                         12350
                         12351
     D307   A9 0E        12352    clrchn  lda  #14
     D309   85 83        12353            sta  sa
     D30B   20 D227      12354    clrc1   jsr  frechn
     D30E   C6 83        12355            dec  sa
     D310   D0 F9        12356            bne  clrc1
     D312   60           12357            rts
                         12358
     D313   A9 0E        12359    cldchn  lda  #14
     D315   85 83        12360            sta  sa
     D317   A6 83        12361    clsd    ldx  sa
     D319   BD 022B      12362            lda  lintab,x
     D31C   C9 FF        12363            cmp  #$ff
     D31E   F0 14        12364            beq  cld2
     D320   29 3F        12365            and  #$3f
     D322   85 82        12366            sta  lindx
     D324   20 DF93      12367            jsr  getact
     D327   AA           12368            tax
     D328   BD 025B      12369            lda  lstjob,x
     D32B   29 01        12370            and  #1
     D32D   C5 7F        12371            cmp  drvnum
     D32F   D0 03        12372            bne  cld2
     D331   20 D227      12373            jsr  frechn
     D334   C6 83        12374    cld2    dec  sa
     D336   10 DF        12375            bpl  clsd
     D338   60           12376            rts
                         12377
                         12378    ;*********************************
                         12379    ;**********  s t l b u f  ******
                         12380    ;*********************************
                         12381
                         12382    ; steal a buffer
                         12383
```

```
Error Addr  Code           Seq    Source statement

                          12384   ; this rtn searches the channels
                          12385   ; in order of least recently used
                          12386   ; and steals the first inactive
                          12387   ; buffer it finds.
                          12388
                          12389   ;     input paramters:
                          12390   ;         lrutbl - least recentyl
                          12391   ;                  used table
                          12392
                          12393   ;     output paramters:
                          12394   ;         a <== buffer #
                          12395
                          12396   ;     register usage:
                          12397   ;         x - channel index
                          12398   ;         y - lrutbl index
                          12399
                          12400   ;*******************************
                          12401
     D339                 12402   stlbuf
     D339   A5 6F         12403           lda  t0
     D33B   48            12404           pha                 ; save t0
     D33C   A0 00         12405           ldy  #0
     D33E                 12406   stl05
     D33E   B6 FA         12407           ldx  lrutbl,y
     D340   B5 A7         12408           lda  buf0,x
     D342   10 04         12409           bpl  stl10          ; active
                          12410
     D344   C9 FF         12411           cmp  #$ff
     D346   D0 16         12412           bne  stl30          ; it's inactive
     D348                 12413   stl10
     D348   8A            12414           txa
     D349   18            12415           clc
     D34A   69 07         12416           adc  #mxchns+1
     D34C   AA            12417           tax
     D34D   B5 A7         12418           lda  buf0,x
     D34F   10 04         12419           bpl  stl20
     D351   C9 FF         12420           cmp  #$ff
     D353   D0 09         12421           bne  stl30
     D355                 12422   stl20
     D355   C8            12423           iny
     D356   C0 05         12424           cpy  #mxchns-1
     D358   90 E4         12425           bcc  stl05
     D35A   A2 FF         12426           ldx  #$ff           ; set failure
     D35C   D0 1C         12427           bne  stl60          ; bra
     D35E                 12428   stl30                       ; steal the buffer if no error
     D35E   86 6F         12429           stx  t0
     D360   29 3F         12430           and  #$3f
     D362   AA            12431           tax
     D363                 12432   stl40
     D363   B5 00         12433           lda  jobs,x
```

```
Error Addr  Code           Seq    Source statement

     D365   30 FC        12434            bmi  stl40       ; wait till done
     D367   C9 02        12435            cmp .#2          ; errors?
     D369   90 08        12436            bcc  stl50       ; ok
     D36B   A6 6F        12437            ldx  t0
     D36D   E0 07        12438            cpx  #mxchns+1
     D36F   90 D7        12439            bcc  stl10       ; check opposite slot
     D371   B0 E2        12440            bcs  stl20       ; check another channel
                         12441
                         12442    ; found one, so lets steal it
                         12443
     D373                12444    stl50
     D373   A4 6F        12445            ldy  t0
     D375   A9 FF        12446            lda  #$ff
     D377   99 00A7      12447            sta  buf0,y      ; clear slot
     D37A                12448    stl60
     D37A   68           12449            pla
     D37B   85 6F        12450            sta  t0
     D37D   8A           12451            txa              ; buf # in .a & set cc's
     D37E   60           12452            rts
                         12453
                         12454
                         12455
                         12456    ; find a free lindx to use
                         12457    ; mark as used in linuse
                         12458
     D37F   A0 00        12459    fndlnx  ldy  #0
     D381   A9 01        12460            lda  #1
     D383   2C 0256      12461    fnd10   bit  linuse      ; 1=free 0=used
     D386   D0 09        12462            bne  fnd30
     D388   C8           12463            iny
     D389   0A           12464            asl  a
     D38A   D0 F7        12465            bne  fnd10
     D38C   A9 70        12466            lda  #nochnl     ; no free lindx available
     D38E   4C C1C8      12467            jmp  cmderr
     D391   49 FF        12468    fnd30   eor  #$ff        ; toggle bit mask
     D393   2D 0256      12469            and  linuse      ; mark bit used
     D396   8D 0256      12470            sta  linuse
     D399   98           12471            tya              ; return lindx in .a
     D39A   60           12472            rts
                         12473            .include  tst4
```

```
Error Addr  Code           Seq    Source statement

                          12475            .subttl 'tst4.src'
                          12476
                          12477    ; get next char from a chanl
                          12478
     D39B                 12479    gbyte                       ; get through internal channel
     D39B   20 D0EB       12480             jsr  fndrch
     D39E   20 C100       12481             jsr  setlds
     D3A1   20 D3AA       12482             jsr  get
     D3A4   A6 82         12483             ldx  lindx
     D3A6   BD 023E       12484             lda  chndat,x
     D3A9   60            12485             rts
                          12486
                          12487
     D3AA   A6 82         12488    get      ldx  lindx
     D3AC   20 D125       12489             jsr  typfil      ; get file type
     D3AF   D0 03         12490             bne  get00
                          12491
     D3B1   4C E120       12492             jmp  rdrel
     D3B4                 12493    get00
     D3B4   A5 83         12494             lda  sa
     D3B6   C9 0F         12495             cmp  #$f         ; cmd channel ?
                          12496
     D3B8   F0 5A         12497             beq  geterc
                          12498
     D3BA   B5 F2         12499             lda  chnrdy,x    ; was last char just sent
     D3BC   29 08         12500             and  #eoisnd     ; just sent eoi
     D3BE   D0 13         12501             bne  get1        ; nope not this time
                          12502
     D3C0   20 D125       12503             jsr  typfil
     D3C3   C9 07         12504             cmp  #dirtyp
     D3C5   D0 07         12505             bne  get0        ; not direct type
                          12506
     D3C7   A9 89         12507             lda  #rndrdy     ;  direct file remains active
     D3C9   95 F2         12508             sta  chnrdy,x    ; talker listener noeoi
     D3CB   4C D3DE       12509             jmp  rndget      ;  prepare the next char
                          12510
     D3CE   A9 00         12511    get0     lda  #notrdy     ; lst char sent,not ready
     D3D0   95 F2         12512             sta  chnrdy,x
     D3D2   60            12513             rts
                          12514
                          12515
     D3D3   A5 83         12516    get1     lda  sa          ; test if a load
     D3D5   F0 32         12517             beq  get6        ; its a load sa of 0
                          12518
     D3D7   20 D125       12519    get2     jsr  typfil      ;  tst for rnd file
     D3DA   C9 04         12520             cmp  #4
     D3DC   90 22         12521             bcc  seqget      ; seq file char get
                          12522
     D3DE   20 D12F       12523    rndget   jsr  getpre      ; direct file get
     D3E1   B5 99         12524             lda  buftab,x
```

```
Error Addr  Code          Seq    Source statement

     D3E3   D9 0244      12525           cmp  lstchr,y    ; up to lst char yet
     D3E6   D0 04        12526           bne  rnget1      ; not yet
                         12527
     D3E8   A9 00        12528           lda  #0          ; read the whole thing
     D3EA   95 99        12529           sta  buftab,x    ; wrap pntr to 0
     D3EC                12530   rnget1
     D3EC   F6 99        12531           inc  buftab,x    ; get the next char
     D3EE                12532   rnget2
     D3EE   A1 99        12533           lda  (buftab,x)
     D3F0                12534   rnget4
     D3F0   99 023E      12535           sta  chndat,y    ; save char in
     D3F3   B5 99        12536           lda  buftab,x
     D3F5   D9 0244      12537           cmp  lstchr,y
     D3F8   D0 05        12538           bne  rnget3
                         12539
     D3FA   A9 81        12540           lda  #rndeoi
     D3FC   99 00F2      12541           sta  chnrdy,y    ; this is last char
     D3FF   60           12542   rnget3  rts              ; send eoi with it
                         12543
                         12544
     D400   20 D156      12545   seqget  jsr  rdbyt      ; read the next byte
     D403   A6 82        12546   get3    ldx  lindx
     D405   9D 023E      12547           sta  chndat,x    ; store in chndat
     D408   60           12548           rts
                         12549
                         12550
     D409   AD 0254      12551   get6    lda  dirlst      ; test if dir lsting
     D40C   F0 F2        12552           beq  seqget      ; no, its not
     D40E   20 ED67      12553           jsr  getdir      ; yes it is,get dir char
     D411   4C D403      12554           jmp  get3
                         12555
     D414                12556   geterc
     D414   20 D4E8      12557           jsr  getpnt
     D417   C9 D4        12558           cmp  #<errbuf-1
     D419   D0 18        12559           bne  ge10
     D41B   A5 95        12560           lda  dirbuf+1
     D41D   C9 02        12561           cmp  #>errbuf
     D41F   D0 12        12562           bne  ge10
                         12563
     D421   A9 0D        12564           lda  #cr
     D423   85 85        12565           sta  data
     D425   20 C123      12566           jsr  erroff
     D428   A9 00        12567           lda  #0
     D42A   20 E6C1      12568           jsr  errts0
     D42D   C6 A5        12569           dec  cb+2
     D42F   A9 80        12570           lda  #eoiout
     D431   D0 12        12571           bne  ge30        ; (jump)
                         12572
     D433                12573   ge10
     D433   20 D137      12574           jsr  getbyt
```

```
Error Addr  Code           Seq    Source statement

     D436   85 85        12575            sta  data
     D438   D0 09        12576            bne  ge20
     D43A                12577   ge15
     D43A   A9 D4        12578            lda  #<errbuf-1
     D43C   20 D4C8      12579            jsr  setpnt
     D43F   A9 02        12580            lda  #>errbuf
     D441   95 9A        12581            sta  buftab+1,x
     D443                12582   ge20
     D443   A9 88        12583            lda  #rdytlk
     D445                12584   ge30
     D445   85 F7        12585            sta  chnrdy+errchn
     D447   A5 85        12586            lda  data
     D449   8D 0243      12587            sta  chndat+errchn
     D44C   60           12588            rts
                         12589
                         12590
                         12591   ; read next buffer of a file
                         12592   ; follow links in first two bytes
                         12593   ; end of file if 1st byte=0
                         12594   ; 2nd char length
                         12595
     D44D   20 DF93      12596   nxtbuf   jsr  getact
     D450   0A           12597            asl  a
     D451   AA           12598            tax
     D452   A9 00        12599            lda  #0
     D454   95 99        12600            sta  buftab,x
     D456   A1 99        12601            lda  (buftab,x)
     D458   F0 05        12602            beq  nxtb1
     D45A   D6 99        12603            dec  buftab,x
     D45C   4C D156      12604            jmp  rdbyt
     D45F   60           12605   nxtb1    rts
                         12606
                         12607   ; direct block read
                         12608
                         12609
     D460   A9 80        12610   drtrd    lda  #read
     D462   D0 02        12611            bne  drt
                         12612
                         12613   ; direct block write
                         12614
     D464   A9 90        12615   drtwrt   lda  #write
     D466   05 7F        12616   drt      ora  drvnum
     D468   8D 024D      12617            sta  cmd
     D46B   A5 F9        12618            lda  jobnum
     D46D   20 D6D3      12619            jsr  seth
     D470   A6 F9        12620            ldx  jobnum
     D472   4C D593      12621            jmp  doit2
                         12622
                         12623   ; open internal read chanl (sa=16)
                         12624
```

```
Error Addr  Code           Seq    Source statement

     D475                12625   opnird
     D475   A9 01        12626           lda  #1
     D477                12627   opntyp
     D477   8D 024A      12628           sta  type
     D47A   A9 11        12629           lda  #irsa
     D47C   85 83        12630           sta  sa
     D47E   20 DC46      12631           jsr  opnrch
     D481   A9 02        12632           lda  #2
     D483   4C D4C8      12633           jmp  setpnt
                         12634
                         12635   ; open internal write chanl (sa=16)
                         12636
     D486   A9 12        12637   opniwr  lda  #iwsa
     D488   85 83        12638           sta  sa
     D48A   4C DCDA      12639           jmp  opnwch
                         12640
                         12641   ; allocate next directory block on 18
                         12642   ;  and mark as used in bam
                         12643
     D48D   20 DE3B      12644   nxdrbk  jsr  curblk
     D490   A9 01        12645           lda  #1
     D492   85 6F        12646           sta  temp
     D494   A5 69        12647           lda  secinc
     D496   48           12648           pha
     D497   A9 03        12649           lda  #3          ;    incr sector by 3 in directory
     D499   85 69        12650           sta  secinc
     D49B   20 F12D      12651           jsr  nxtds
     D49E   68           12652           pla
     D49F   85 69        12653           sta  secinc
     D4A1   A9 00        12654           lda  #0
     D4A3   20 D4C8      12655           jsr  setpnt
     D4A6   A5 80        12656           lda  track
     D4A8   20 CFF1      12657           jsr  putbyt
     D4AB   A5 81        12658           lda  sector
     D4AD   20 CFF1      12659           jsr  putbyt
     D4B0   20 D0C7      12660           jsr  wrtbuf
     D4B3   20 D599      12661           jsr  watjob
     D4B6   A9 00        12662           lda  #0
     D4B8   20 D4C8      12663           jsr  setpnt
     D4BB   20 CFF1      12664   nxdb1   jsr  putbyt
     D4BE   D0 FB        12665           bne  nxdb1
     D4C0   20 CFF1      12666           jsr  putbyt
     D4C3   A9 FF        12667           lda  #$ff
     D4C5   4C CFF1      12668           jmp  putbyt
                         12669
                         12670
                         12671   ; .a=new pntr value
                         12672
     D4C8   85 6F        12673   setpnt  sta  temp
     D4CA   20 DF93      12674           jsr  getact
```

```
Error Addr  Code           Seq    Source statement

     D4CD   OA           12675            asl  a
     D4CE   AA           12676            tax
     D4CF   B5 9A        12677            lda  buftab+1,x
     D4D1   85 95        12678            sta  dirbuf+1
     D4D3   A5 6F        12679            lda  temp
     D4D5   95 99        12680            sta  buftab,x
     D4D7   85 94        12681            sta  dirbuf
     D4D9   60           12682            rts
                         12683
                         12684
                         12685    ; free the internal chanl (sa=16)
                         12686
     D4DA   A9 11        12687    freich  lda  #irsa
     D4DC   85 83        12688            sta  sa
     D4DE   20 D227      12689            jsr  frechn
     D4E1   A9 12        12690            lda  #iwsa
     D4E3   85 83        12691            sta  sa
     D4E5   4C D227      12692            jmp  frechn
                         12693
                         12694    ; read the active buffer pointer
                         12695
     D4E8   20 DF93      12696    getpnt  jsr  getact
     D4EB                12697    setdir
     D4EB   OA           12698    gp1     asl  a
     D4EC   AA           12699            tax
     D4ED   B5 9A        12700            lda  buftab+1,x
     D4EF   85 95        12701            sta  dirbuf+1
     D4F1   B5 99        12702            lda  buftab,x
     D4F3   85 94        12703            sta  dirbuf
     D4F5   60           12704            rts
                         12705
                         12706    ; direct read byte, .a=byte# to read
                         12707
     D4F6   85 71        12708    drdbyt  sta  temp+2
     D4F8   20 DF93      12709            jsr  getact
     D4FB   AA           12710            tax
     D4FC   BD FEEO      12711            lda  bufind,x
     D4FF   85 72        12712            sta  temp+3
     D501   AO OO        12713            ldy  #0
     D503   B1 71        12714            lda  (temp+2),y
     D505   60           12715            rts
                         12716
                         12717    ; index table containing high
                         12718    ; byte address of buffers
                         12719            .include  jobssf
```

```
Error Addr  Code           Seq    Source statement

                          12721           .subttl 'jobssf.src'
                          12722
                          12723   ; use lastjob for drive #
                          12724   ; cmd is used for job command
                          12725
     D506                 12726   setljb
     D506   BD 025B       12727           lda  lstjob,x
     D509   29 01         12728           and  #1
     D50B   0D 024D       12729           ora  cmd
                          12730
                          12731   ; set job up and check t&s
                          12732   ;  .a=command for jobs
                          12733   ;  .x=job number
                          12734
     D50E                 12735   setjob
     D50E   48            12736           pha
     D50F   86 F9         12737           stx  jobnum
     D511   8A            12738           txa
     D512   0A            12739           asl  a
     D513   AA            12740           tax
     D514   B5 07         12741           lda  hdrs+1,x    ; 4/12**********
     D516   8D 024D       12742           sta  cmd         ; save sector
     D519   B5 06         12743           lda  hdrs,x      ; 4/12**********
     D51B   F0 2D         12744           beq  tserr
                          12745
     D51D   CD 02AC       12746           cmp  maxtrk
     D520   B0 28         12747           bcs  tserr       ; track too large
                          12748
     D522   AA            12749           tax
     D523   68            12750           pla              ; check for write
     D524   48            12751           pha
     D525   29 F0         12752           and  #$f0
     D527   C9 90         12753           cmp  #write
     D529   D0 4F         12754           bne  sjb1        ; not write,skip check
     D52B   68            12755           pla
     D52C   48            12756           pha
     D52D   4A            12757           lsr  a
     D52E   B0 05         12758           bcs  sjb2        ; drive 1
                          12759
     D530   AD 0101       12760           lda  dskver      ; get version #
     D533   90 03         12761           bcc  sjb3
     D535                 12762   sjb2
     D535   AD 0102       12763           lda  dskver+1    ; get drive 1 ver#
     D538                 12764   sjb3
     D538   F0 05         12765           beq  sjb4        ; no # is ok, too
     D53A   CD FED5       12766           cmp  vernum
     D53D   D0 33         12767           bne  vnerr       ; not same vernum #
                          12768
     D53F                 12769   sjb4
     D53F   8A            12770           txa              ; restore track #
```

```
Error Addr  Code          Seq    Source statement

     D540   20 F24B      12771            jsr  maxsec
     D543   CD 024D      12772            cmp  cmd
     D546   F0 02        12773            beq  tserr
     D548   B0 30        12774            bcs  sjb1       ; sector is ok!
                         12775
                         12776
                         12777    ; illegal track and sector
                         12778
     D54A                12779    tserr
     D54A   20 D552      12780            jsr  hed2ts
     D54D                12781    tser1
     D54D   A9 66        12782            lda  #badts
     D54F   4C E645      12783            jmp  cmder2
                         12784
     D552                12785    hed2ts
     D552   A5 F9        12786            lda  jobnum
     D554   0A           12787            asl  a
     D555   AA           12788            tax
     D556   B5 06        12789            lda  hdrs,x      ; 4/12***********
     D558   85 80        12790            sta  track
     D55A   B5 07        12791            lda  hdrs+1,x    ; 4/12***********
     D55C   85 81        12792            sta  sector
     D55E   60           12793            rts
                         12794
                         12795
     D55F                12796    tschk
     D55F   A5 80        12797            lda  track
     D561   F0 EA        12798            beq  tser1
     D563   CD 02AC      12799            cmp  maxtrk
     D566   B0 E5        12800            bcs  tser1
                         12801
     D568   20 F24B      12802            jsr  maxsec
     D56B   C5 81        12803            cmp  sector
     D56D   F0 DE        12804            beq  tser1
     D56F   90 DC        12805            bcc  tser1
     D571   60           12806            rts
                         12807
     D572                12808    vnerr
     D572   20 D552      12809            jsr  hed2ts
     D575   A9 73        12810            lda  #cbmv2      ; write to wrong version
     D577   4C E645      12811            jmp  cmder2
                         12812
     D57A                12813    sjb1
     D57A   A6 F9        12814            ldx  jobnum
     D57C   68           12815            pla
     D57D   8D 024D      12816            sta  cmd
     D580   95 00        12817            sta  jobs,x
     D582   9D 025B      12818            sta  lstjob,x
     D585   60           12819            rts
                         12820
```

```
Error Addr  Code           Seq    Source statement

                          12821
                          12822   ; do job in .a, set up error count
                          12823   ; and lstjob. return when job done ok
                          12824   ; jmp to error if error returns
                          12825
     D586                 12826   doread
     D586   A9 80         12827           lda  #read
     D588   D0 02         12828           bne  dojob      ; bra
     D58A                 12829   dowrit
     D58A   A9 90         12830           lda  #write
     D58C                 12831   dojob
     D58C   05 7F         12832           ora  drvnum
     D58E   A6 F9         12833           ldx  jobnum
                          12834
     D590   8D 024D       12835   doit    sta  cmd
     D593   AD 024D       12836   doit2   lda  cmd
     D596   20 D50E       12837           jsr  setjob
                          12838
                          12839   ; wait until job(.x) is done
                          12840   ; return when done
                          12841
     D599   20 D5A6       12842   watjob  jsr  tstjob
     D59C   B0 FB         12843           bcs  watjob
     D59E   48            12844           pha             ; clr jobrtn flag
     D59F   A9 00         12845           lda  #0
     D5A1   8D 0298       12846           sta  jobrtn
     D5A4   68            12847           pla
     D5A5   60            12848           rts
                          12849
                          12850
                          12851   ;test if job(.x) is done yet
                          12852   ;if not done return
                          12853   ;if ok then return else redo it
                          12854
     D5A6   B5 00         12855   tstjob  lda  jobs,x
     D5A8   30 1A         12856           bmi  notyet
     D5AA   C9 02         12857           cmp  #2
     D5AC   90 14         12858           bcc  ok
                          12859
     D5AE   C9 08         12860           cmp  #8         ; check for wp switch on
     D5B0   F0 08         12861           beq  tj10
                          12862                             .
     D5B2   C9 0B         12863           cmp  #11        ; check for id mismatch
     D5B4   F0 04         12864           beq  tj10
                          12865
     D5B6   C9 0F         12866           cmp  #$f        ; check for nodrive
     D5B8   D0 0C         12867           bne  recov
     D5BA   2C 0298       12868   tj10    bit  jobrtn
     D5BD   30 03         12869           bmi  ok
     D5BF   4C D63F       12870           jmp  quit2
```

```
Error Addr  Code           Seq    Source statement

                          12871
     D5C2   18            12872  ok      clc             ; c=0 finished ok or quit
     D5C3   60            12873          rts
                          12874
     D5C4   38            12875  notyet  sec             ; c=1 not yet
     D5C5   60            12876          rts
                          12877
                          12878
     D5C6                 12879  recov
     D5C6   98            12880          tya             ; save .y
     D5C7   48            12881          pha
     D5C8   A5 7F         12882          lda  drvnum     ; save drive #
     D5CA   48            12883          pha
     D5CB   BD 025B       12884          lda  lstjob,x
     D5CE   29 01         12885          and  #1
     D5D0   85 7F         12886          sta  drvnum     ; set active drive #
                          12887
     D5D2   A8            12888          tay
     D5D3   B9 FECA       12889          lda  ledmsk,y
     D5D6   8D 026D       12890          sta  erled
                          12891
     D5D9   20 D6A6       12892          jsr  dorec
     D5DC   C9 02         12893          cmp  #2
     D5DE   B0 03         12894          bcs  rec01
     D5E0   4C D66D       12895          jmp  rec95
     D5E3                 12896  rec01
                          12897
     D5E3   BD 025B       12898          lda  lstjob,x   ; original job
     D5E6   29 F0         12899          and  #$f0       ; mask job code
     D5E8   48            12900          pha             ; save it
     D5E9   C9 90         12901          cmp  #write
     D5EB   D0 07         12902          bne  rec0       ; not a write
                          12903
     D5ED   A5 7F         12904          lda  drvnum
     D5EF   09 B8         12905          ora  #secsek    ; replace w/ sector seek...
     D5F1   9D 025B       12906          sta  lstjob,x   ; ... during recovery
     D5F4                 12907  rec0
     D5F4   24 6A         12908          bit  revcnt
     D5F6   70 39         12909          bvs  rec5       ; no track offset
                          12910
     D5F8   A9 00         12911          lda  #0
     D5FA   8D 0299       12912          sta  eptr       ; clr offset table ptr
     D5FD   8D 029A       12913          sta  toff       ; clr total offset
     D600                 12914  rec1
     D600   AC 0299       12915          ldy  eptr
     D603   AD 029A       12916          lda  toff
     D606   38            12917          sec
     D607   F9 FEDB       12918          sbc  offset,y
     D60A   8D 029A       12919          sta  toff       ; keep track of all offsets
                          12920
```

```
Error Addr  Code          Seq    Source statement

     D60D   B9 FEDB      12921            lda  offset,y
                         12922
     D610   20 FFA1      12923            jsr  ptch12     ; set micro-stepping flag *rom-05ds 01/21/85*
                         12924    ;       jsr  hedoff
                         12925
     D613   EE 0299      12926            inc  eptr       ; bump table ptr
     D616   20 D6A6      12927            jsr  dorec      ; do the recovery
     D619   C9 02        12928            cmp  #2         ; error code < 2?
     D61B   90 08        12929            bcc  rec3       ; job worked
                         12930
     D61D   AC 0299      12931            ldy  eptr
     D620   B9 FEDB      12932            lda  offset,y
     D623   D0 DB        12933            bne  rec1       ; null indicates end
     D625                12934    rec3
     D625   AD 029A      12935            lda  toff
                         12936
     D628   20 FFA6      12937            jsr  ptch13     ; clr micro-stepping flag *rom-05ds 01/21/85*
                         12938    ;       jsr  hedoff
                         12939
     D62B   B5 00        12940            lda  jobs,x
     D62D   C9 02        12941            cmp  #2
     D62F   90 2B        12942            bcc  rec9       ; no error
     D631                12943    rec5
     D631   24 6A        12944            bit  revcnt     ; check bump-on flag
     D633   10 0F        12945            bpl  rec7       ; no bump
                         12946
     D635                12947    quit
     D635   68           12948            pla
     D636   C9 90        12949            cmp  #write     ; check original job
     D638   D0 05        12950            bne  quit2
                         12951
     D63A   05 7F        12952            ora  drvnum
     D63C   9D 025B      12953            sta  lstjob,x   ; must restore original
     D63F                12954    quit2
     D63F   B5 00        12955            lda  jobs,x     ; .a= error #
     D641   20 E60A      12956            jsr  error
     D644                12957    rec7
     D644   68           12958            pla
     D645   2C 0298      12959            bit  jobrtn
     D648   30 23        12960            bmi  rec95      ; return job error
     D64A   48           12961            pha
                         12962
                         12963    ;do the bump
                         12964
     D64B   A9 C0        12965            lda  #bump
     D64D   05 7F        12966            ora  drvnum
     D64F   95 00        12967            sta  jobs,x
     D651                12968    rec8
                         12969    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     D651   20 9FB6      12971            jsr  stbctl      ; strobe the controller
                         12972                             ; ***rom ds 01/22/85***
     D654   EA            12973            nop
                         12974    ;       lda  jobs,x
                         12975    ;       bmi  rec8        ; wait
                         12976
                         12977    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     D693                13021   movhed
     D693   48           13022           pha             ; save .a
     D694   98           13023           tya             ; put phase in .a
     D695   A4 7F        13024           ldy  drvnum
     D697   99 02FE      13025           sta  phase,y
     D69A                13026   mh10
     D69A   D9 02FE      13027           cmp  phase,y
     D69D   F0 FB        13028           beq  mh10       ; wait for controller to change it
                         13029
     D69F   A9 00        13030           lda  #0
     D6A1   99 02FE      13031           sta  phase,y    ; clear it out
     D6A4   68           13032           pla             ; restore
     D6A5   60           13033           rts
                         13034
                         13035
     D6A6                13036   dorec                   ; do last job recovery
     D6A6   A5 6A        13037           lda  revcnt     ; re-try job revcnt...
     D6A8   29 3F        13038           and  #$3f       ; ...# of times
     D6AA   A8           13039           tay
     D6AB                13040   dorec1
     D6AB   AD 026D      13041           lda  erled
     D6AE   4D 1C00      13042           eor  ledprt
     D6B1   8D 1C00      13043           sta  ledprt
     D6B4   BD 025B      13044           lda  lstjob,x   ; set last job
     D6B7   95 00        13045           sta  jobs,x
     D6B9                13046   dorec2
     D6B9   20 9FB6      13047           jsr  stbctl     ; strobe the controller *rom-05ds 01/22/85*
     D6BC   EA           13048           nop
                         13049   ;       lda  jobs,x     ; wait
                         13050   ;       bmi  dorec2
                         13051
     D6BD   C9 02        13052           cmp  #2
     D6BF   90 03        13053           bcc  dorec3     ; it worked
                         13054
     D6C1   88           13055           dey
     D6C2   D0 E7        13056           bne  dorec1     ; keep trying
     D6C4                13057   dorec3
     D6C4   48           13058           pha
     D6C5   AD 026D      13059           lda  erled      ; leave drive led on
     D6C8   0D 1C00      13060           ora  ledprt
     D6CB   8D 1C00      13061           sta  ledprt
     D6CE   68           13062           pla
     D6CF   60           13063           rts             ; finished
                         13064
                         13065   ; set header of active buffer of the
                         13066   ; current lindx to track,sector,id
                         13067
     D6D0   20 DF93      13068   sethdr  jsr  getact
     D6D3                13069   seth
     D6D3   0A           13070           asl  a
```

```
Error Addr  Code           Seq    Source statement

     D6D4   A8           13071           tay
     D6D5   A5 80        13072           lda  track
     D6D7   99 0006      13073           sta  hdrs,y      ; set track
     D6DA   A5 81        13074           lda  sector
     D6DC   99 0007      13075           sta  hdrs+1,y    ; set sector
     D6DF   A5 7F        13076           lda  drvnum      ; get proper id(drvnum)
     D6E1   0A           13077           asl  a
     D6E2   AA           13078           tax
     D6E3   60           13079           rts
                         13080           .include  addfil
```

```
Error Addr  Code           Seq    Source statement

                          13082           .subttl 'addfil.src'
                          13083   ; add file to directory
                          13084
     D6E4   A5 83         13085   addfil  lda  sa          ; save variables
     D6E6   48            13086           pha
     D6E7   A5 82         13087           lda  lindx
     D6E9   48            13088           pha
     D6EA   A5 81         13089           lda  sector
     D6EC   48            13090           pha
     D6ED   A5 80         13091           lda  track
     D6EF   48            13092           pha
     D6F0   A9 11         13093           lda  #irsa
     D6F2   85 83         13094           sta  sa
     D6F4   20 DE3B       13095           jsr  curblk     ; use last accessed search
     D6F7   AD 024A       13096           lda  type
     D6FA   48            13097           pha
     D6FB   A5 E2         13098           lda  fildrv
     D6FD   29 01         13099           and  #1
     D6FF   85 7F         13100           sta  drvnum
     D701   A6 F9         13101           ldx  jobnum
     D703   5D 025B       13102           eor  lstjob,x
     D706   4A            13103           lsr  a
     D707   90 0C         13104           bcc  af08       ; same drive as required
                          13105
     D709   A2 01         13106           ldx  #1
     D70B   8E 0292       13107           stx  delind     ; look for deleted entry
     D70E   20 C5AC       13108           jsr  srchst
     D711   F0 1D         13109           beq  af15       ; all full, new sector
     D713   D0 28         13110           bne  af20       ; found one
                          13111
     D715   AD 0291       13112   af08    lda  delsec
     D718   F0 0C         13113           beq  af10       ; deleted entry not located
     D71A   C5 81         13114           cmp  sector
     D71C   F0 1F         13115           beq  af20       ; sector is resident
     D71E   85 81         13116           sta  sector
     D720   20 D460       13117           jsr  drtrd      ; read sector in
     D723   4C D73D       13118           jmp  af20
                          13119
     D726   A9 01         13120   af10    lda  #1         ; find deleted entry
     D728   8D 0292       13121           sta  delind
     D72B   20 C617       13122           jsr  search
     D72E   D0 0D         13123           bne  af20
     D730   20 D48D       13124   af15    jsr  nxdrbk     ; all full, new sector
     D733   A5 81         13125           lda  sector
     D735   8D 0291       13126           sta  delsec
     D738   A9 02         13127           lda  #2
     D73A   8D 0292       13128           sta  delind
     D73D   AD 0292       13129   af20    lda  delind
     D740   20 D4C8       13130           jsr  setpnt
     D743   68            13131           pla
```

```
Error Addr  Code           Seq    Source statement

     D744   8D 024A      13132            sta  type        ; set type
     D747   C9 04        13133            cmp  #reltyp
     D749   D0 02        13134            bne  af25
     D74B   09 80        13135            ora  #$80
     D74D                13136  af25
     D74D   20 CFF1      13137            jsr  putbyt
     D750   68           13138            pla
     D751   8D 0280      13139            sta  filtrk      ; ...table & entry
     D754   20 CFF1      13140            jsr  putbyt
     D757   68           13141            pla
     D758   8D 0285      13142            sta  filsec      ; set sector link in...
     D75B   20 CFF1      13143            jsr  putbyt      ; ...table & entry
     D75E   20 DF93      13144            jsr  getact
     D761   A8           13145            tay
     D762   AD 027A      13146            lda  filtbl
     D765   AA           13147            tax
     D766   A9 10        13148            lda  #16
     D768   20 C66E      13149            jsr  trname      ; transfer name
     D76B   A0 10        13150            ldy  #16
     D76D   A9 00        13151            lda  #0          ; clear # of blocks &...
     D76F   91 94        13152  af30      sta  (dirbuf),y ; ...& replace links
     D771   C8           13153            iny
     D772   C0 1B        13154            cpy  #27
     D774   90 F9        13155            bcc  af30
     D776   AD 024A      13156            lda  type        ; a relative file ?
     D779   C9 04        13157            cmp  #reltyp
     D77B   D0 13        13158            bne  af50        ; no
     D77D   A0 10        13159            ldy  #16         ; yes
     D77F   AD 0259      13160            lda  trkss       ; get ss track
     D782   91 94        13161            sta  (dirbuf),y ; put in directory
     D784   C8           13162            iny
     D785   AD 025A      13163            lda  secss       ; get ss sector
     D788   91 94        13164            sta  (dirbuf),y ; put in
     D78A   C8           13165            iny
     D78B   AD 0258      13166            lda  rec         ; get record size
     D78E   91 94        13167            sta  (dirbuf),y
     D790   20 D464      13168  af50      jsr  drtwrt      ; write it out
     D793   68           13169            pla
     D794   85 82        13170            sta  lindx
     D796   AA           13171            tax
     D797   68           13172            pla
     D798   85 83        13173            sta  sa
     D79A   AD 0291      13174            lda  delsec
     D79D   85 D8        13175            sta  entsec
     D79F   9D 0260      13176            sta  dsec,x
     D7A2   AD 0292      13177            lda  delind
     D7A5   85 DD        13178            sta  entind
     D7A7   9D 0266      13179            sta  dind,x
     D7AA   AD 024A      13180            lda  type
     D7AD   85 E7        13181            sta  pattyp
```

```
Error Addr  Code           Seq    Source statement

     D7AF   A5 7F        13182           lda  drvnum
     D7B1   85 E2        13183           sta  fildrv
     D7B3   60           13184           rts
                         13185           .include  open
```

```
Error Addr  Code           Seq    Source statement

                          13187           .subttl 'open.src'
                          13188
                          13189   ;  open channel from ieee
                          13190   ;  parses the input string that is
                          13191   ;  sent as an open data channel,
                          13192   ;  load, or save.  channels are allocated
                          13193   ;  and the directory is searched for
                          13194   ;  the filename contained in the string.
                          13195
     D7B4                 13196   open
     D7B4   A5 83         13197           lda  sa
     D7B6   8D 024C       13198           sta  tempsa
     D7B9   20 C2B3       13199           jsr  cmdset     ; initiate cmd ptrs
     D7BC   8E 022A       13200           stx  cmdnum
     D7BF   AE 0200       13201           ldx  cmdbuf
     D7C2   AD 024C       13202           lda  tempsa
     D7C5   D0 2C         13203           bne  op021
     D7C7   E0 2A         13204           cpx  #'*        ; load last?
     D7C9   D0 28         13205           bne  op021
     D7CB   A5 7E         13206           lda  prgtrk
     D7CD   F0 4D         13207           beq  op0415     ; no last prog, init 0
                          13208
     D7CF                 13209   op02                    ; load last program
     D7CF   85 80         13210           sta  track
     D7D1   AD 026E       13211           lda  prgdrv
     D7D4   85 7F         13212           sta  drvnum
     D7D6   85 E2         13213           sta  fildrv
     D7D8   A9 02         13214           lda  #prgtyp
     D7DA   85 E7         13215           sta  pattyp
     D7DC   AD 026F       13216           lda  prgsec
     D7DF   85 81         13217           sta  sector
     D7E1   20 C100       13218           jsr  setlds     ;  make sure led gets turned on!!
     D7E4   20 DC46       13219           jsr  opnrch
     D7E7   A9 04         13220           lda  #prgtyp+prgtyp
     D7E9   05 7F         13221           ora  drvnum
     D7EB   A6 82         13222   endrd   ldx  lindx
     D7ED   99 00EC       13223           sta  filtyp,y
     D7F0   4C C194       13224           jmp  endcmd
                          13225
     D7F3   E0 24         13226   op021   cpx  #'$
     D7F5   D0 1E         13227           bne  op041
     D7F7   AD 024C       13228           lda  tempsa     ; load directory
     D7FA   D0 03         13229           bne  op04
     D7FC   4C DA55       13230           jmp  loadir
                          13231
     D7FF   20 C1D1       13232   op04    jsr  simprs     ; open dir as seq file
     D802   AD FE85       13233           lda  dirtrk
     D805   85 80         13234           sta  track
     D807   A9 00         13235           lda  #0
     D809   85 81         13236           sta  sector
```

```
Error Addr  Code          Seq    Source statement

     D80B   20 DC46      13237           jsr  opnrch
     D80E   A5 7F        13238           lda  drvnum
     D810   09 02        13239           ora  #seqtyp+seqtyp
     D812   4C D7EB      13240           jmp  endrd
                         13241
     D815   E0 23        13242   op041   cpx  #'#         ; "#" is direct access
     D817   D0 12        13243           bne  op042
     D819   4C CB84      13244           jmp  opnblk
                         13245
     D81C   A9 02        13246   op0415  lda  #prgtyp     ; program type
     D81E   8D 0296      13247           sta  typflg
     D821   A9 00        13248           lda  #0
     D823   85 7F        13249           sta  drvnum
     D825   8D 028E      13250           sta  lstdrv
     D828   20 D042      13251           jsr  initdr
                         13252
     D82B                13253   op042
     D82B   20 C1E5      13254           jsr  prscln      ; look for ":"
     D82E   D0 04        13255           bne  op049
     D830   A2 00        13256           ldx  #0
     D832   F0 0C        13257           beq  op20        ; bra
     D834                13258   op049
     D834   8A           13259           txa
     D835   F0 05        13260           beq  op10
     D837   A9 30        13261   op05    lda  #badsyn     ; something amiss
     D839   4C C1C8      13262           jmp  cmderr
     D83C   88           13263   op10    dey              ; back up to ":"
     D83D   F0 01        13264           beq  op20        ; 1st char is ":"
     D83F   88           13265           dey
     D840   8C 027A      13266   op20    sty  filtbl      ; save filename ptr
     D843   A9 8D        13267           lda  #$8d        ; look for cr-shifted
     D845   20 C268      13268           jsr  parse
                         13269
     D848   E8           13270           inx
     D849   8E 0278      13271           stx  f2cnt
     D84C   20 C312      13272           jsr  onedrv
     D84F   20 C3CA      13273           jsr  optsch
     D852   20 C49D      13274           jsr  ffst        ; look for file entry
     D855   A2 00        13275           ldx  #0
     D857   8E 0258      13276           stx  rec
     D85A   8E 0297      13277           stx  mode        ; read mode
     D85D   8E 024A      13278           stx  type        ; deleted
     D860   E8           13279           inx
     D861   EC 0277      13280           cpx  f1cnt
     D864   B0 10        13281           bcs  op40        ; no parameters
                         13282
     D866   20 DA09      13283           jsr  cktm        ; check for type & mode
     D869   E8           13284           inx
     D86A   EC 0277      13285           cpx  f1cnt
     D86D   B0 07        13286           bcs  op40        ; only one parameter
```

```
Error Addr  Code           Seq    Source statement

                          13287
      D86F  CO 04         13288            cpy  #reltyp
      D871  FO 3E         13289            beq  op60        ; set record size
                          13290
      D873  20 DA09       13291            jsr  cktm        ; set type/mode
      D876                13292  op40
      D876  AE 024C       13293            ldx  tempsa
      D879  86 83         13294            stx  sa          ; set sa back
      D87B  EO 02         13295            cpx  #2
      D87D  BO 12         13296            bcs  op45        ; not load or save
                          13297
      D87F  8E 0297       13298            stx  mode        ; mode=sa
      D882  A9 40         13299            lda  #$40
      D884  8D 02F9       13300            sta  wbam
      D887  AD 024A       13301            lda  type
      D88A  DO 1B         13302            bne  op50        ; type from parm
                          13303
      D88C  A9 02         13304            lda  #prgtyp
      D88E  8D 024A       13305            sta  type        ; use prg
      D891                13306  op45
      D891  AD 024A       13307            lda  type
      D894  DO 11         13308            bne  op50        ; type from parm
                          13309
      D896  A5 E7         13310            lda  pattyp
      D898  29 07         13311            and  #typmsk
      D89A  8D 024A       13312            sta  type        ; type from file
                          13313
      D89D  AD 0280       13314            lda  filtrk
      D8A0  DO 05         13315            bne  op50        ; yes, it exists
                          13316
      D8A2  A9 01         13317            lda  #seqtyp
      D8A4  8D 024A       13318            sta  type        ; default is seq
      D8A7                13319  op50
      D8A7  AD 0297       13320            lda  mode
      D8AA  C9 01         13321            cmp  #wtmode
      D8AC  FO 18         13322            beq  op75        ; go write
                          13323
      D8AE  4C D940       13324            jmp  op90
                          13325
      D8B1                13326  op60
      D8B1  BC 027A       13327            ldy  filtbl,x    ; get record size
      D8B4  B9 0200       13328            lda  cmdbuf,y
      D8B7  8D 0258       13329            sta  rec
      D8BA  AD 0280       13330            lda  filtrk
      D8BD  DO B7         13331            bne  op40        ; it's here, read
                          13332
      D8BF  A9 01         13333            lda  #wtmode     ; use write to open
      D8C1  8D 0297       13334            sta  mode
      D8C4  DO BO         13335            bne  op40        ; (bra)
                          13336
```

```
Error Addr  Code           Seq    Source statement

     D8C6                13337   op75
     D8C6   A5 E7        13338           lda  pattyp
     D8C8   29 80        13339           and  #$80
     D8CA   AA           13340           tax
     D8CB   D0 14        13341           bne  op81
     D8CD   A9 20        13342   op77    lda  #$20       ; open write
     D8CF   24 E7        13343           bit  pattyp
     D8D1   F0 06        13344           beq  op80
     D8D3   20 C8B6      13345           jsr  deldir     ; created
     D8D6   4C D9E3      13346           jmp  opwrit
                         13347
     D8D9   AD 0280      13348   op80    lda  filtrk
     D8DC   D0 03        13349           bne  op81
     D8DE   4C D9E3      13350           jmp  opwrit     ; not found, ok!
     D8E1   AD 0200      13351   op81    lda  cmdbuf
     D8E4   C9 40        13352           cmp  #'@        ; check for replace
     D8E6   F0 0D        13353           beq  op82
     D8E8   8A           13354           txa
     D8E9   D0 05        13355           bne  op815
     D8EB   A9 63        13356           lda  #flexst
     D8ED   4C C1C8      13357           jmp  cmderr
     D8F0                13358   op815
     D8F0   A9 33        13359           lda  #badfn
     D8F2   4C C1C8      13360           jmp  cmderr
                         13361
                         13362   ;********* check for bug here******
     D8F5                13363   op82
     D8F5   A5 E7        13364           lda  pattyp     ; replace
     D8F7   29 07        13365           and  #typmsk
     D8F9   CD 024A      13366           cmp  type
     D8FC   D0 67        13367           bne  op115
     D8FE   C9 04        13368           cmp  #reltyp
     D900   F0 63        13369           beq  op115
                         13370
     D902   20 DCDA      13371           jsr  opnwch
     D905   A5 82        13372           lda  lindx
     D907   8D 0270      13373           sta  wlindx
     D90A   A9 11        13374           lda  #irsa      ; internal chan
     D90C   85 83        13375           sta  sa
     D90E   20 D0EB      13376           jsr  fndrch
     D911   AD 0294      13377           lda  index
     D914   20 D4C8      13378           jsr  setpnt
     D917   A0 00        13379           ldy  #0
     D919   B1 94        13380           lda  (dirbuf),y
     D91B   09 20        13381           ora  #$20       ; set replace bit
     D91D   91 94        13382           sta  (dirbuf),y
                         13383
     D91F   A0 1A        13384           ldy  #26
     D921   A5 80        13385           lda  track
     D923   91 94        13386           sta  (dirbuf),y
```

```
Error Addr  Code           Seq    Source statement

     D925   C8            13387            iny
     D926   A5 81         13388            lda  sector
     D928   91 94         13389            sta  (dirbuf),y
                          13390
     D92A   AE 0270       13391            ldx  wlindx
     D92D   A5 D8         13392            lda  entsec
     D92F   9D 0260       13393            sta  dsec,x
     D932   A5 DD         13394            lda  entind
     D934   9D 0266       13395            sta  dind,x
     D937   20 DE3B       13396            jsr  curblk
     D93A   20 D464       13397            jsr  drtwrt
     D93D   4C D9EF       13398            jmp  opfin
                          13399
                          13400  ;*********************************
                          13401
                          13402
     D940   AD 0280       13403  op90      lda  filtrk      ; open read (& load)
     D943   D0 05         13404             bne  op100
     D945                 13405  op95
     D945   A9 62         13406             lda  #flntfd     ; track not recorded
     D947   4C C1C8       13407             jmp  cmderr      ; not found
     D94A                 13408  op100
     D94A   AD 0297       13409             lda  mode
     D94D   C9 03         13410             cmp  #mdmode
     D94F   F0 0B         13411             beq  op110
     D951   A9 20         13412             lda  #$20
     D953   24 E7         13413             bit  pattyp
     D955   F0 05         13414             beq  op110
     D957   A9 60         13415             lda  #filopn
     D959   4C C1C8       13416             jmp  cmderr
     D95C   A5 E7         13417  op110     lda  pattyp
     D95E   29 07         13418             and  #typmsk     ; type is in index table
     D960   CD 024A       13419             cmp  type
     D963   F0 05         13420             beq  op120
     D965   A9 64         13421  op115     lda  #mistyp     ; type mismatch
     D967   4C C1C8       13422             jmp  cmderr
     D96A                 13423  op120                      ; everything is ok!
     D96A   A0 00         13424             ldy  #0
     D96C   8C 0279       13425             sty  f2ptr
     D96F   AE 0297       13426             ldx  mode
     D972   E0 02         13427             cpx  #apmode
     D974   D0 1A         13428             bne  op125
     D976   C9 04         13429             cmp  #reltyp
     D978   F0 EB         13430             beq  op115
                          13431
     D97A   B1 94         13432             lda  (dirbuf),y
     D97C   29 4F         13433             and  #$4f
     D97E   91 94         13434             sta  (dirbuf),y
     D980   A5 83         13435             lda  sa
     D982   48            13436             pha
```

```
Error Addr  Code           Seq    Source statement

     D983   A9 11        13437            lda  #irsa
     D985   85 83        13438            sta  sa
     D987   20 DE3B      13439            jsr  curblk
     D98A   20 D464      13440            jsr  drtwrt
     D98D   68           13441            pla
     D98E   85 83        13442            sta  sa
     D990                13443   op125
     D990   20 D9A0      13444            jsr  opread
     D993   AD 0297      13445            lda  mode
     D996   C9 02        13446            cmp  #apmode
     D998   D0 55        13447            bne  opfin
                         13448
     D99A   20 DA2A      13449            jsr  append
     D99D   4C C194      13450            jmp  endcmd
                         13451
                         13452   ;**********************
     D9A0                13453   opread
     D9A0   A0 13        13454            ldy  #19
     D9A2   B1 94        13455            lda  (dirbuf),y
     D9A4   8D 0259      13456            sta  trkss
     D9A7   C8           13457            iny
     D9A8   B1 94        13458            lda  (dirbuf),y
     D9AA   8D 025A      13459            sta  secss
     D9AD   C8           13460            iny
     D9AE   B1 94        13461            lda  (dirbuf),y
     D9B0   AE 0258      13462            ldx  rec
     D9B3   8D 0258      13463            sta  rec
     D9B6   8A           13464            txa
     D9B7   F0 0A        13465            beq  op130
     D9B9   CD 0258      13466            cmp  rec
     D9BC   F0 05        13467            beq  op130
     D9BE   A9 50        13468            lda  #norec
     D9C0   20 C1C8      13469            jsr  cmderr
     D9C3                13470   op130
     D9C3   AE 0279      13471            ldx  f2ptr
     D9C6   BD 0280      13472            lda  filtrk,x
     D9C9   85 80        13473            sta  track
     D9CB   BD 0285      13474            lda  filsec,x
     D9CE   85 81        13475            sta  sector
     D9D0   20 DC46      13476            jsr  opnrch
     D9D3   A4 82        13477            ldy  lindx       ; open a read chnl
     D9D5   AE 0279      13478            ldx  f2ptr
     D9D8   B5 D8        13479            lda  entsec,x
     D9DA   99 0260      13480            sta  dsec,y
     D9DD   B5 DD        13481            lda  entind,x
     D9DF   99 0266      13482            sta  dind,y
     D9E2   60           13483            rts
                         13484
     D9E3                13485   opwrit
     D9E3   A5 E2        13486            lda  fildrv
```

```
Error Addr  Code          Seq    Source statement

     D9E5   29 01        13487            and  #1
     D9E7   85 7F        13488            sta  drvnum
     D9E9   20 DCDA      13489            jsr  opnwch
     D9EC   20 D6E4      13490            jsr  addfil      ; add to directory
     D9EF                13491  opfin
     D9EF   A5 83        13492            lda  sa
     D9F1   C9 02        13493            cmp  #2
     D9F3   B0 11        13494            bcs  opf1
                         13495
     D9F5   20 DE3E      13496            jsr  gethdr
     D9F8   A5 80        13497            lda  track
     D9FA   85 7E        13498            sta  prgtrk
                         13499
     D9FC   A5 7F        13500            lda  drvnum
     D9FE   8D 026E      13501            sta  prgdrv
                         13502
     DA01   A5 81        13503            lda  sector
     DA03   8D 026F      13504            sta  prgsec
     DA06                13505  opf1
     DA06   4C C199      13506            jmp  endsav
                         13507
     DA09                13508  cktm
     DA09   BC 027A      13509            ldy  filtbl,x    ; get ptr
     DA0C   B9 0200      13510            lda  cmdbuf,y    ; get char
     DA0F   A0 04        13511            ldy  #nmodes
     DA11                13512  ckm1
     DA11   88           13513            dey
     DA12   30 08        13514            bmi  ckm2        ; no valid mode
                         13515
     DA14   D9 FEB2      13516            cmp  modlst,y
     DA17   D0 F8        13517            bne  ckm1
     DA19   8C 0297      13518            sty  mode        ; mode found
     DA1C                13519  ckm2
     DA1C   A0 05        13520            ldy  #ntypes
     DA1E                13521  ckt1
     DA1E   88           13522            dey
     DA1F   30 08        13523            bmi  ckt2        ; no valid type
                         13524
     DA21   D9 FEB6      13525            cmp  tplst,y
     DA24   D0 F8        13526            bne  ckt1
     DA26   8C 024A      13527            sty  type        ; type found
     DA29                13528  ckt2
     DA29   60           13529            rts
                         13530
     DA2A                13531  append
     DA2A   20 CA39      13532            jsr  gcbyte
     DA2D   A9 80        13533            lda  #lrf
     DA2F   20 DDA6      13534            jsr  tstflg
     DA32   F0 F6        13535            beq  append
                         13536
```

```
Error Addr  Code           Seq    Source statement

     DA34   20 DE95      13537            jsr  rdlnk
     DA37   A6 81        13538            ldx  sector
     DA39   E8           13539            inx
     DA3A   8A           13540            txa
     DA3B   D0 05        13541            bne  ap30
     DA3D   20 D1A3      13542            jsr  wrt0        ; get another block
     DA40   A9 02        13543            lda  #2
     DA42                13544   ap30
     DA42   20 D4C8      13545            jsr  setpnt
     DA45   A6 82        13546            ldx  lindx
     DA47   A9 01        13547            lda  #rdylst
     DA49   95 F2        13548            sta  chnrdy,x
     DA4B   A9 80        13549            lda  #$80        ; chnl bit
     DA4D   05 82        13550            ora  lindx
     DA4F   A6 83        13551            ldx  sa
     DA51   9D 022B      13552            sta  lintab,x
     DA54   60           13553            rts
                         13554
                         13555
                         13556   ; load directory
     DA55                13557   loadir
     DA55   A9 0C        13558            lda  #ldcmd
     DA57   8D 022A      13559            sta  cmdnum
     DA5A   A9 00        13560            lda  #0          ; load only drive zero
     DA5C   AE 0274      13561            ldx  cmdsiz
     DA5F   CA           13562            dex
     DA60   F0 0B        13563            beq  ld02
                         13564
     DA62   CA           13565   ld01     dex              ; load by name
     DA63   D0 21        13566            bne  ld03
     DA65   AD 0201      13567            lda  cmdbuf+1
     DA68   20 C3BD      13568            jsr  tst0v1
     DA6B   30 19        13569            bmi  ld03
                         13570
     DA6D                13571   ld02                      ; load dir with a star
     DA6D   85 E2        13572            sta  fildrv
     DA6F   EE 0277      13573            inc  f1cnt
     DA72   EE 0278      13574            inc  f2cnt
     DA75   EE 027A      13575            inc  filtbl
     DA78   A9 80        13576            lda  #$80
     DA7A   85 E7        13577            sta  pattyp
     DA7C   A9 2A        13578            lda  #'*
     DA7E   8D 0200      13579            sta  cmdbuf      ; cover both cases
     DA81   8D 0201      13580            sta  cmdbuf+1
     DA84   D0 18        13581            bne  ld10        ; (branch)
                         13582
     DA86                13583   ld03
     DA86   20 C1E5      13584            jsr  prscln
     DA89   D0 05        13585            bne  ld05        ; found ":"
                         13586   ;search by name on both drives
```

```
Error Addr  Code          Seq    Source statement

     DA8B   20 C2DC      13587            jsr  cmdrst
     DA8E   A0 03        13588            ldy  #3
     DA90   88           13589    ld05    dey
     DA91   88           13590            dey
     DA92   8C 027A      13591            sty  filtbl
                         13592
     DA95   20 C200      13593            jsr  tc35       ; parse & set tables
     DA98   20 C398      13594            jsr  fs1set
     DA9B   20 C320      13595            jsr  alldrs
                         13596
     DA9E   20 C3CA      13597    ld10    jsr  optsch     ; new directory
     DAA1   20 C7B7      13598            jsr  newdir
     DAA4   20 C49D      13599            jsr  ffst
     DAA7   20 EC9E      13600    ld20    jsr  stdir      ; start directory
     DAAA   20 D137      13601            jsr  getbyt     ; set 1st byte
     DAAD   A6 82        13602            ldx  lindx
     DAAF   9D 023E      13603            sta  chndat,x
     DAB2   A5 7F        13604            lda  drvnum
     DAB4   8D 028E      13605            sta  lstdrv
     DAB7   09 04        13606            ora  #4
     DAB9   95 EC        13607            sta  filtyp,x
     DABB   A9 00        13608            lda  #0
     DABD   85 A3        13609            sta  buftab+cbptr
     DABF   60           13610            rts
                         13611            .include  close
```

```
Error Addr  Code           Seq    Source statement

                          13613           .subttl 'close.src'
                          13614   ; close the file associated with sa
                          13615
     DAC0                 13616   close
     DAC0   A9 00         13617           lda  #0
     DAC2   8D 02F9       13618           sta  wbam
     DAC5   A5 83         13619           lda  sa
     DAC7   D0 0B         13620           bne  cls10      ; directory close
     DAC9   A9 00         13621           lda  #0
     DACB   8D 0254       13622           sta  dirlst     ; clear dir list
     DACE   20 D227       13623           jsr  frechn
     DAD1                 13624   cls05
     DAD1   4C D4DA       13625           jmp  freich
     DAD4                 13626   cls10
     DAD4   C9 0F         13627           cmp  #$f
     DAD6   F0 14         13628           beq  clsall     ; close cmd chanl
     DAD8   20 DB02       13629           jsr  clschn     ; close channel
     DADB   A5 83         13630           lda  sa
     DADD   C9 02         13631           cmp  #2
     DADF   90 F0         13632           bcc  cls05
                          13633
     DAE1   AD 026C       13634           lda  erword
     DAE4   D0 03         13635           bne  cls15      ; last command had an error
     DAE6   4C C194       13636           jmp  endcmd
     DAE9                 13637   cls15
     DAE9   4C C1AD       13638           jmp  scren1
                          13639
     DAEC                 13640   clsall
     DAEC   A9 0E         13641           lda  #14
     DAEE   85 83         13642           sta  sa
     DAF0                 13643   cls20
     DAF0   20 DB02       13644           jsr  clschn
     DAF3   C6 83         13645           dec  sa
     DAF5   10 F9         13646           bpl  cls20
     DAF7   AD 026C       13647           lda  erword
     DAFA   D0 03         13648           bne  cls25      ;  last command had an error
     DAFC   4C C194       13649           jmp  endcmd
     DAFF                 13650   cls25
     DAFF   4C C1AD       13651           jmp  scren1
                          13652
     DB02                 13653   clschn
     DB02   A6 83         13654           ldx  sa
     DB04   BD 022B       13655           lda  lintab,x
     DB07   C9 FF         13656           cmp  #$ff
     DB09   D0 01         13657           bne  clsc28
     DB0B   60            13658           rts
     DB0C                 13659   clsc28
     DB0C   29 0F         13660           and  #$f
     DB0E   85 82         13661           sta  lindx
                          13662
```

```
Error Addr  Code          Seq    Source statement

     DB10   20 D125      13663            jsr  typfil
     DB13   C9 07        13664            cmp  #dirtyp
     DB15   F0 0F        13665            beq  clsc30     ; direct channel
     DB17   C9 04        13666            cmp  #reltyp
     DB19   F0 11        13667            beq  clsrel
                         13668
     DB1B   20 D107      13669            jsr  fndwch     ; look for write channel
     DB1E   B0 09        13670            bcs  clsc31
                         13671
     DB20   20 DB62      13672            jsr  clswrt     ; close seq write
     DB23   20 DBA5      13673            jsr  clsdir     ; close directory
     DB26                13674   clsc30
     DB26   20 EEF4      13675            jsr  mapout     ; write bam
     DB29                13676   clsc31
     DB29   4C D227      13677            jmp  frechn
                         13678
     DB2C                13679   clsrel
     DB2C   20 DDF1      13680            jsr  scrub
     DB2F   20 CF1E      13681            jsr  dblbuf
     DB32   20 E1CB      13682            jsr  ssend
     DB35   A6 D5        13683            ldx  ssnum
     DB37   86 73        13684            stx  t4
     DB39   E6 73        13685            inc  t4
     DB3B   A9 00        13686            lda  #0
     DB3D   85 70        13687            sta  t1
     DB3F   85 71        13688            sta  t2
     DB41   A5 D6        13689            lda  ssind
     DB43   38           13690            sec
     DB44   E9 0E        13691            sbc  #ssioff-2
     DB46   85 72        13692            sta  t3
     DB48   20 DF51      13693            jsr  sscalc
     DB4B   A6 82        13694            ldx  lindx
     DB4D   A5 70        13695            lda  t1
     DB4F   95 B5        13696            sta  nbkl,x
     DB51   A5 71        13697            lda  t2
     DB53   95 BB        13698            sta  nbkh,x
     DB55   A9 40        13699            lda  #dyfile
     DB57   20 DDA6      13700            jsr  tstflg
     DB5A   F0 03        13701            beq  clsr1
     DB5C   20 DBA5      13702            jsr  clsdir
     DB5F   4C D227      13703   clsr1    jmp  frechn
                         13704
                         13705   ; close a write chanl
                         13706
     DB62                13707   clswrt                   ; close seq write file
     DB62   A6 82        13708            ldx  lindx
     DB64   B5 B5        13709            lda  nbkl,x
     DB66   15 BB        13710            ora  nbkh,x
     DB68   D0 0C        13711            bne  clsw10     ; at least 1 block written
                         13712
```

```
Error Addr  Code           Seq    Source statement

     DB6A   20 D4E8       13713            jsr  getpnt
     DB6D   C9 02         13714            cmp  #2
     DB6F   D0 05         13715            bne  clsw10     ; at least 1 byte written
                          13716
     DB71   A9 0D         13717            lda  #cr
     DB73   20 CFF1       13718            jsr  putbyt
     DB76                 13719   clsw10
     DB76   20 D4E8       13720            jsr  getpnt
     DB79   C9 02         13721            cmp  #2
     DB7B   D0 0F         13722            bne  clsw20     ; not mt buffer
                          13723
     DB7D   20 CF1E       13724            jsr  dblbuf     ; switch bufs
                          13725
     DB80   A6 82         13726            ldx  lindx
     DB82   B5 B5         13727            lda  nbkl,x
     DB84   D0 02         13728            bne  clsw15
     DB86   D6 BB         13729            dec  nbkh,x
     DB88                 13730   clsw15
     DB88   D6 B5         13731            dec  nbkl,x
                          13732
     DB8A   A9 00         13733            lda  #0
     DB8C                 13734   clsw20
     DB8C   38            13735            sec
     DB8D   E9 01         13736            sbc  #1         ; back up 1
     DB8F   48            13737            pha             ; save it
     DB90   A9 00         13738            lda  #0
     DB92   20 D4C8       13739            jsr  setpnt
     DB95   20 CFF1       13740            jsr  putbyt     ; tlink=0
     DB98   68            13741            pla             ; lstchr count
     DB99   20 CFF1       13742            jsr  putbyt
                          13743
     DB9C   20 D0C7       13744            jsr  wrtbuf     ; write out last buffer
     DB9F   20 D599       13745            jsr  watjob     ; finish job up
     DBA2   4C CF1E       13746            jmp  dblbuf     ; make sure both bufs ok
                          13747
                          13748   ; directory close on open write file
                          13749
     DBA5   A6 82         13750   clsdir   ldx  lindx      ; save lindx
     DBA7   8E 0270       13751            stx  wlindx     ; &sa
     DBAA   A5 83         13752            lda  sa
     DBAC   48            13753            pha
     DBAD   BD 0260       13754            lda  dsec,x     ; get directory sector
     DBB0   85 81         13755            sta  sector
     DBB2   BD 0266       13756            lda  dind,x     ; get sector offset
     DBB5   8D 0294       13757            sta  index
     DBB8   B5 EC         13758            lda  filtyp,x   ; drv # in filtyp
     DBBA   29 01         13759            and  #1
     DBBC   85 7F         13760            sta  drvnum
     DBBE   AD FE85       13761            lda  dirtrk
     DBC1   85 80         13762            sta  track
```

```
Error Addr  Code           Seq    Source statement

     DBC3   20 DF93      13763           jsr  getact      ; allocate a buffer
     DBC6   48           13764           pha
     DBC7   85 F9        13765           sta  jobnum
     DBC9   20 D460      13766           jsr  drtrd       ; read directory sector
     DBCC   A0 00        13767           ldy  #0
     DBCE   BD FEE0      13768           lda  bufind,x    ; .x is job#
     DBD1   85 87        13769           sta  r0+1
     DBD3   AD 0294      13770           lda  index
     DBD6   85 86        13771           sta  r0
     DBD8   B1 86        13772           lda  (r0),y
     DBDA   29 20        13773           and  #$20
     DBDC   F0 43        13774           beq  clsd5
     DBDE   20 D125      13775           jsr  typfil
     DBE1   C9 04        13776           cmp  #reltyp
     DBE3   F0 44        13777           beq  clsd6
                         13778
     DBE5   B1 86        13779           lda  (r0),y
     DBE7   29 8F        13780           and  #$8f        ; replace file
     DBE9   91 86        13781           sta  (r0),y
     DBEB   C8           13782           iny
     DBEC   B1 86        13783           lda  (r0),y
     DBEE   85 80        13784           sta  track
     DBF0   84 71        13785           sty  temp+2
     DBF2   A0 1B        13786           ldy  #27         ; extract replacement link
     DBF4   B1 86        13787           lda  (r0),y      ;  to last sector
     DBF6   48           13788           pha
     DBF7   88           13789           dey
     DBF8   B1 86        13790           lda  (r0),y
     DBFA   D0 0A        13791           bne  clsd4
     DBFC   85 80        13792           sta  track
     DBFE   68           13793           pla
     DBFF   85 81        13794           sta  sector
     DC01   A9 67        13795           lda  #$67
     DC03   20 E645      13796           jsr  cmder2
     DC06                13797   clsd4
     DC06   48           13798           pha
     DC07   A9 00        13799           lda  #0
     DC09   91 86        13800           sta  (r0),y
     DC0B   C8           13801           iny
     DC0C   91 86        13802           sta  (r0),y
     DC0E   68           13803           pla
     DC0F   A4 71        13804           ldy  temp+2
     DC11   91 86        13805           sta  (r0),y
     DC13   C8           13806           iny
     DC14   B1 86        13807           lda  (r0),y
     DC16   85 81        13808           sta  sector
     DC18   68           13809           pla
     DC19   91 86        13810           sta  (r0),y
     DC1B   20 C87D      13811           jsr  delfil      ; delete old file
     DC1E   4C DC29      13812           jmp  clsd6       ; set close bit
```

```
Error Addr  Code           Seq    Source statement

     DC21                13813   clsd5
     DC21   B1 86        13814           lda  (r0),y
     DC23   29 0F        13815           and  #$f
     DC25   09 80        13816           ora  #$80
     DC27   91 86        13817           sta  (r0),y
     DC29   AE 0270      13818   clsd6   ldx  wlindx
     DC2C   A0 1C        13819           ldy  #28        ; set # of blocks
     DC2E   B5 B5        13820           lda  nbkl,x
     DC30   91 86        13821           sta  (r0),y
     DC32   C8           13822           iny
     DC33   B5 BB        13823           lda  nbkh,x
     DC35   91 86        13824           sta  (r0),y
     DC37   68           13825           pla
     DC38   AA           13826           tax
     DC39   A9 90        13827           lda  #write     ; write directory sector
     DC3B   05 7F        13828           ora  drvnum
     DC3D   20 D590      13829           jsr  doit
     DC40   68           13830           pla
     DC41   85 83        13831           sta  sa
     DC43   4C D107      13832           jmp  fndwch     ; restore lindx
                         13833           .include  opnchnl
```

```
Error Addr  Code           Seq    Source statement

                          13835           .subttl 'opnchnl.src'
                          13836   ; opnchnl
                          13837
                          13838   ; open a read chanl with 2 buffers
                          13839   ; will insert sa in lintab
                          13840   ; and inits all pointers.
                          13841   ; relative ss and ptrs are set.
                          13842
     DC46   A9 01         13843   opnrch  lda  #1         ; get one data buffer
     DC48   20 D1E2       13844           jsr  getrch
     DC4B   20 DCB6       13845           jsr  initp      ; clear pointers
     DC4E   AD 024A       13846           lda  type
     DC51   48            13847           pha
     DC52   0A            13848           asl  a
     DC53   05 7F         13849           ora  drvnum
     DC55   95 EC         13850           sta  filtyp,x   ; set file type
     DC57   20 D09B       13851           jsr  strrd      ; read 1st one or two blocks
     DC5A   A6 82         13852           ldx  lindx
     DC5C   A5 80         13853           lda  track
     DC5E   D0 05         13854           bne  or10
                          13855
     DC60   A5 81         13856           lda  sector
     DC62   9D 0244       13857           sta  lstchr,x   ; set last char ptr
     DC65                 13858   or10
     DC65   68            13859           pla
     DC66   C9 04         13860           cmp  #reltyp
     DC68   D0 3F         13861           bne  or30       ; must be sequential stuff
                          13862
     DC6A   A4 83         13863           ldy  sa
     DC6C   B9 022B       13864           lda  lintab,y   ; set channel as r/w
     DC6F   09 40         13865           ora  #$40
     DC71   99 022B       13866           sta  lintab,y
                          13867
     DC74   AD 0258       13868           lda  rec
     DC77   95 C7         13869           sta  rs,x       ; set record size
                          13870
     DC79   20 D28E       13871           jsr  getbuf     ; get ss buffer
     DC7C   10 03         13872           bpl  or20
     DC7E   4C D20F       13873           jmp  gberr      ; no buffer
     DC81                 13874   or20
     DC81   A6 82         13875           ldx  lindx
     DC83   95 CD         13876           sta  ss,x
     DC85   AC 0259       13877           ldy  trkss      ; set ss track
     DC88   84 80         13878           sty  track
     DC8A   AC 025A       13879           ldy  secss      ; set ss sector
     DC8D   84 81         13880           sty  sector
     DC8F   20 D6D3       13881           jsr  seth       ; set ss header
     DC92   20 DE73       13882           jsr  rdss       ; read it in
     DC95   20 D599       13883           jsr  watjob
     DC98                 13884   orow
```

```
Error Addr  Code           Seq    Source statement

     DC98   A6 82        13885            ldx  lindx
     DC9A   A9 02        13886            lda  #2
     DC9C   95 C1        13887            sta  nr,x        ; set for nxtrec
                         13888
     DC9E   A9 00        13889            lda  #0
     DCA0   20 D4C8      13890            jsr  setpnt      ; set first data byte
                         13891
     DCA3   20 E153      13892            jsr  rd40        ; set up 1st record
     DCA6   4C DE3E      13893            jmp  gethdr      ; restore t&s
                         13894
     DCA9                13895   or30
     DCA9   20 D156      13896            jsr  rdbyt       ; sequential set up
     DCAC   A6 82        13897            ldx  lindx
     DCAE   9D 023E      13898            sta  chndat,x
     DCB1   A9 88        13899            lda  #rdytlk
     DCB3   95 F2        13900            sta  chnrdy,x
     DCB5   60           13901            rts
                         13902
                         13903   ; initialize variables for open chanl
                         13904   ; lstjob,sets active buffer#,lstchr,
                         13905   ; buffer pointers in buftab=2
                         13906
                         13907   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          13935   ;       sta  lstchr,x
                          13936   ;       rts
                          13937
                          13938   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     DD2A   A6 82        13985            ldx  lindx
     DD2C   B5 CD        13986            lda  ss,x
     DD2E   20 D6D3      13987            jsr  seth        ; set ss header
     DD31   A9 00        13988            lda  #0
     DD33   20 DEE9      13989            jsr  setssp
     DD36   A9 00        13990            lda  #0          ; set null link
     DD38   20 DD8D      13991            jsr  putss
     DD3B   A9 11        13992            lda  #ssioff+1   ; set last char
     DD3D   20 DD8D      13993            jsr  putss
     DD40   A9 00        13994            lda  #0          ; set this ss #
     DD42   20 DD8D      13995            jsr  putss
     DD45   AD 0258      13996            lda  rec         ; record size
     DD48   20 DD8D      13997            jsr  putss
     DD4B   A5 80        13998            lda  track
     DD4D   20 DD8D      13999            jsr  putss
     DD50   A5 81        14000            lda  sector
     DD52   20 DD8D      14001            jsr  putss
     DD55   A9 10        14002            lda  #ssioff
     DD57   20 DEE9      14003            jsr  setssp
     DD5A   20 DE3E      14004            jsr  gethdr      ; get first t&s
     DD5D   A5 80        14005            lda  track
     DD5F   20 DD8D      14006            jsr  putss
     DD62   A5 81        14007            lda  sector
     DD64   20 DD8D      14008            jsr  putss
                         14009
     DD67   20 DE6C      14010            jsr  wrtss       ; write it out
     DD6A   20 D599      14011            jsr  watjob
     DD6D   A9 02        14012            lda  #2
     DD6F   20 D4C8      14013            jsr  setpnt
                         14014
     DD72   A6 82        14015            ldx  lindx       ; set nr for null buffer
     DD74   38           14016            sec
     DD75   A9 00        14017            lda  #0
     DD77   F5 C7        14018            sbc  rs,x
     DD79   95 C1        14019            sta  nr,x
                         14020
     DD7B   20 E2E2      14021            jsr  nulbuf      ; null records
     DD7E   20 DE19      14022            jsr  nullnk
     DD81   20 DE5E      14023            jsr  wrtout
     DD84   20 D599      14024            jsr  watjob
     DD87   20 EEF4      14025            jsr  mapout
     DD8A   4C DC98      14026            jmp  orow
                         14027
                         14028   ;*
                         14029   ;*
                         14030   ;***********************
                         14031   ;*
                         14032   ;* putss
                         14033   ;*
                         14034   ;* put byte into side sector
```

```
Error Addr  Code           Seq    Source statement

                         14035    ;*
                         14036    ;************************
                         14037    ;*
                         14038    ;*
     DD8D   48           14039    putss   pha
     DD8E   A6 82        14040            ldx  lindx
     DD90   B5 CD        14041            lda  ss,x
     DD92   4C CFFD      14042            jmp  putb1
                         14043            .include  tstflg
```

```
Error Addr  Code           Seq    Source statement

                          14045           .subttl 'tstflg.src'
                          14046   ;*
                          14047   ;*
                          14048   ;**********************************
                          14049   ;*
                          14050   ;* scflg
                          14051   ;* setflg
                          14052   ;* clrflg
                          14053   ;* tstflg
                          14054   ;*
                          14055   ;**********************************
                          14056   ;*
                          14057   ;*
     DD95   90 06         14058   scflg   bcc  clrflg
                          14059
     DD97   A6 82         14060   setflg  ldx  lindx
     DD99   15 EC         14061           ora  filtyp,x
     DD9B   D0 06         14062           bne  clrf10
                          14063
     DD9D   A6 82         14064   clrflg  ldx  lindx
     DD9F   49 FF         14065           eor  #$ff
     DDA1   35 EC         14066           and  filtyp,x
     DDA3   95 EC         14067   clrf10  sta  filtyp,x
     DDA5   60            14068           rts
                          14069
                          14070
     DDA6   A6 82         14071   tstflg  ldx  lindx
     DDA8   35 EC         14072           and  filtyp,x
     DDAA   60            14073           rts
                          14074
                          14075   ;*
                          14076   ;*
                          14077   ;******************************
                          14078   ;*
                          14079   ;*
                          14080   ;*  tstwrt
                          14081   ;*
                          14082   ;******************************
                          14083   ;*
                          14084   ;*
     DDAB   20 DF93       14085   tstwrt  jsr  getact
     DDAE   AA            14086           tax
     DDAF   BD 025B       14087           lda  lstjob,x
     DDB2   29 F0         14088           and  #$f0
     DDB4   C9 90         14089           cmp  #$90
     DDB6   60            14090           rts
                          14091   ;*
                          14092   ;*
```

```
Error Addr  Code           Seq    Source statement

                          14094
                          14095   ; test for active files from
                          14096   ; lindx table
                          14097
                          14098   ; c=1 file not active x=18,y=?,a=?
                          14099   ; c=0 file active x=entfnd,y=lindx,a=?
                          14100
                          14101
     DDB7   A2 00         14102   tstchn  ldx  #0          ; start search at top
     DDB9   86 71         14103   tstc20  stx  temp+2      ; save to look on
     DDBB   BD 022B       14104           lda  lintab,x    ; get lindx
     DDBE   C9 FF         14105           cmp  #$ff
     DDC0   D0 08         14106           bne  tstc40      ; if plus test it
     DDC2   A6 71         14107   tstc30  ldx  temp+2      ; not active
     DDC4   E8            14108           inx
     DDC5   E0 10         14109           cpx  #maxsa-2    ; searched all
     DDC7   90 F0         14110           bcc  tstc20      ; no
     DDC9   60            14111   tstrts  rts              ; yes
                          14112
     DDCA   86 71         14113   tstc40  stx  temp+2      ; save x
     DDCC   29 3F         14114           and  #$3f
     DDCE   A8            14115           tay              ; use lindx as index
     DDCF   B9 00EC       14116           lda  filtyp,y    ; right drive # ?
     DDD2   29 01         14117           and  #1
     DDD4   85 70         14118           sta  temp+1
     DDD6   AE 0253       14119           ldx  entfnd      ; index entry found
     DDD9   B5 E2         14120           lda  fildrv,x
     DDDB   29 01         14121           and  #1
     DDDD   C5 70         14122           cmp  temp+1      ; same drive # ?
     DDDF   D0 E1         14123           bne  tstc30      ; no
     DDE1   B9 0260       14124           lda  dsec,y      ; yes - same dir. entry ?
     DDE4   D5 D8         14125           cmp  entsec,x
     DDE6   D0 DA         14126           bne  tstc30      ; no
     DDE8   B9 0266       14127           lda  dind,y
     DDEB   D5 DD         14128           cmp  entind,x
     DDED   D0 D3         14129           bne  tstc30      ; no
     DDEF   18            14130           clc              ; set flag
     DDF0   60            14131           rts
                          14132           .include  tsutil
```

```
Error Addr  Code           Seq    Source statement

                          14134            .subttl 'tsutil.src'
                          14135   ;*
                          14136   ;*
                          14137   ;*********************************
                          14138   ;*
                          14139   ;*  scrub
                          14140   ;*
                          14141   ;*  write out buffer if dirty
                          14142   ;*
                          14143   ;*********************************
                          14144   ;*
                          14145   ;*
     DDF1                 14146   scrub
     DDF1   20 DF9E       14147           jsr  gaflgs
     DDF4   50 06         14148           bvc  scr1       ; not dirty
                          14149
     DDF6   20 DE5E       14150           jsr  wrtout
     DDF9   20 D599       14151           jsr  watjob
     DDFC   60            14152   scr1    rts
                          14153   ;*
                          14154   ;*
                          14155   ;*********************************
                          14156   ;*
                          14157   ;*  setlnk
                          14158   ;*
                          14159   ;*  put track,sector into buffer
                          14160   ;*
                          14161   ;*********************************
                          14162   ;*
                          14163   ;*
     DDFD   20 DE2B       14164   setlnk  jsr  set00
                          14165
     DE00   A5 80         14166           lda  track
     DE02   91 94         14167           sta  (dirbuf),y
     DE04   C8            14168           iny
     DE05   A5 81         14169           lda  sector
     DE07   91 94         14170           sta  (dirbuf),y
     DE09   4C E105       14171           jmp  sdirty
                          14172   ;*
                          14173   ;*
                          14174   ;*********************************
                          14175   ;*
                          14176   ;*  getlnk
                          14177   ;*
                          14178   ;*  get link from buffer into
                          14179   ;*  track and sector
                          14180   ;*
                          14181   ;*********************************
                          14182   ;*
                          14183   ;*
```

```
Error Addr  Code           Seq    Source statement

     DE0C   20 DE2B      14184   getlnk  jsr  set00
                         14185
     DE0F   B1 94        14186           lda  (dirbuf),y
     DE11   85 80        14187           sta  track
     DE13   C8           14188           iny
     DE14   B1 94        14189           lda  (dirbuf),y
     DE16   85 81        14190           sta  sector
     DE18   60           14191           rts
                         14192   ;*
                         14193   ;*
                         14194   ;*********************************
                         14195   ;*
                         14196   ;*  nullnk
                         14197   ;*
                         14198   ;*  set track link=0 & sector
                         14199   ;*  link=last non-zero char.
                         14200   ;*
                         14201   ;*********************************
                         14202   ;*
                         14203   ;*
     DE19                14204   nullnk
     DE19   20 DE2B      14205           jsr  set00
     DE1C   A9 00        14206           lda  #0
     DE1E   91 94        14207           sta  (dirbuf),y
     DE20   C8           14208           iny
     DE21   A6 82        14209           ldx  lindx
     DE23   B5 C1        14210           lda  nr,x
     DE25   AA           14211           tax
     DE26   CA           14212           dex
     DE27   8A           14213           txa
     DE28   91 94        14214           sta  (dirbuf),y
     DE2A   60           14215           rts
                         14216   ;*
                         14217   ;*
                         14218   ;*******************************
                         14219   ;*
                         14220   ;* set00
                         14221   ;*
                         14222   ;* setup up pointer to buffer
                         14223   ;*
                         14224   ;*******************************
                         14225   ;*
                         14226   ;*
     DE2B   20 DF93      14227   set00   jsr  getact
     DE2E   0A           14228           asl  a
     DE2F   AA           14229           tax
     DE30   B5 9A        14230           lda  buftab+1,x
     DE32   85 95        14231           sta  dirbuf+1
     DE34   A9 00        14232           lda  #0
     DE36   85 94        14233           sta  dirbuf
```

```
Error Addr  Code           Seq    Source statement

     DE38   A0 00        14234            ldy  #0
     DE3A   60            14235            rts
                          14236   ;*
                          14237   ;*
                          14238   ;*******************************
                          14239   ;*
                          14240   ;*  gethdr
                          14241   ;*
                          14242   ;*  read track,setor from header
                          14243   ;*
                          14244   ;*******************************
                          14245   ;*
                          14246   ;*
     DE3B   20 D0EB       14247   curblk  jsr  fndrch
     DE3E   20 DF93       14248   gethdr  jsr  getact
     DE41   85 F9         14249            sta  jobnum
     DE43   0A            14250            asl  a
     DE44   A8            14251            tay
     DE45   B9 0006       14252            lda  hdrs,y      ; 4/12**************
     DE48   85 80         14253            sta  track
     DE4A   B9 0007       14254            lda  hdrs+1,y    ; 4/12**************
     DE4D   85 81         14255            sta  sector
     DE4F   60            14256            rts
                          14257   ;*
                          14258   ;*
                          14259   ;*******************************
                          14260   ;*
                          14261   ;* wrtab,rdab  wrtout,rdin
                          14262   ;* wrtss,rdss
                          14263   ;*
                          14264   ;*******************************
                          14265   ;*
                          14266   ;*
     DE50   A9 90         14267   wrtab   lda  #write
     DE52   8D 024D       14268            sta  cmd
     DE55   D0 28         14269            bne  sj10
                          14270
     DE57   A9 80         14271   rdab    lda  #read
     DE59   8D 024D       14272            sta  cmd
     DE5C   D0 21         14273            bne  sj10
                          14274
     DE5E   A9 90         14275   wrtout  lda  #write
     DE60   8D 024D       14276            sta  cmd
     DE63   D0 26         14277            bne  sj20
                          14278
     DE65   A9 80         14279   rdin    lda  #read
     DE67   8D 024D       14280            sta  cmd
     DE6A   D0 1F         14281            bne  sj20
                          14282
     DE6C   A9 90         14283   wrtss   lda  #write
```

```
Error Addr  Code           Seq    Source statement

     DE6E   8D 024D      14284            sta  cmd
     DE71   D0 02        14285            bne  rds5
                         14286
     DE73   A9 80        14287    rdss    lda  #read
     DE75   8D 024D      14288    rds5    sta  cmd
     DE78   A6 82        14289            ldx  lindx
     DE7A   B5 CD        14290            lda  ss,x
     DE7C   AA           14291            tax
     DE7D   10 13        14292            bpl  sj30       ; was...bne sj30
                         14293
     DE7F   20 D6D0      14294    sj10    jsr  sethdr
     DE82   20 DF93      14295            jsr  getact
     DE85   AA           14296            tax
     DE86   A5 7F        14297            lda  drvnum
     DE88   9D 025B      14298            sta  lstjob,x
     DE8B   20 E115      14299    sj20    jsr  cdirty
     DE8E   20 DF93      14300            jsr  getact
     DE91   AA           14301            tax
     DE92   4C D506      14302    sj30    jmp  setljb
                         14303    ;*
                         14304    ;*
                         14305    ;*
                         14306    ;***************************
                         14307    ;*
                         14308    ;*      rdlnk
                         14309    ;*
                         14310    ;***************************
                         14311    ;*
                         14312    ;*
     DE95   A9 00        14313    rdlnk   lda  #0
     DE97   20 D4C8      14314            jsr  setpnt
     DE9A   20 D137      14315            jsr  getbyt
     DE9D   85 80        14316            sta  track
     DE9F   20 D137      14317            jsr  getbyt
     DEA2   85 81        14318            sta  sector
     DEA4   60           14319            rts
                         14320            .include  ssutil
```

```
Error Addr  Code           Seq    Source statement

                          14322            .subttl 'ssutil.src'
                          14323
                          14324    ;*********************************
                          14325    ;* bOtobO: transfer bytes from   *
                          14326    ;*          one buf to other.    *
                          14327    ;*    reg: in: .a= # bytes       *
                          14328    ;*             .y= source buf #  *
                          14329    ;*             .x= destin buf #  *
                          14330    ;*********************************
                          14331
     DEA5                 14332    bOtobO
     DEA5   48            14333             pha
     DEA6   A9 OO         14334             lda  #O
     DEA8   85 6F         14335             sta  temp
     DEAA   85 71         14336             sta  temp+2
     DEAC   B9 FEEO       14337             lda  bufind,y
     DEAF   85 7O         14338             sta  temp+1
     DEB1   BD FEEO       14339             lda  bufind,x
     DEB4   85 72         14340             sta  temp+3
     DEB6   68            14341             pla
     DEB7   A8            14342             tay
     DEB8   88            14343             dey
     DEB9                 14344    bO2
     DEB9   B1 6F         14345             lda  (temp),y
     DEBB   91 71         14346             sta  (temp+2),y
     DEBD   88            14347             dey
     DEBE   1O F9         14348             bpl  bO2
     DECO   6O            14349             rts
                          14350
                          14351    ;*********************************
                          14352    ;* clrbuf: clear buffer given    *
                          14353    ;*    reg: in: .a= buffer #      *
                          14354    ;*         out: .y,.a =O         *
                          14355    ;*********************************
                          14356
     DEC1                 14357    clrbuf
     DEC1   A8            14358             tay
     DEC2   B9 FEEO       14359             lda  bufind,y
     DEC5   85 7O         14360             sta  temp+1
     DEC7   A9 OO         14361             lda  #O
     DEC9   85 6F         14362             sta  temp
     DECB   A8            14363             tay
     DECC                 14364    cb1O
     DECC   91 6F         14365             sta  (temp),y
     DECE   C8            14366             iny
     DECF   DO FB         14367             bne  cb1O
     DED1   6O            14368             rts
                          14369
                          14370    ;*********************************
                          14371    ;* ssset: set ss pntr to O       *
```

```
Error Addr  Code           Seq    Source statement

                          14372   ;*   reg: out: .a= ss number     *
                          14373   ;*********************************
                          14374
     DED2                 14375   ssset
     DED2   A9 00         14376           lda  #0
     DED4   20 DEDC       14377           jsr  ssdir
     DED7   A0 02         14378           ldy  #2
     DED9   B1 94         14379           lda  (dirbuf),y
     DEDB   60            14380           rts
                          14381
                          14382   ;*********************************
                          14383   ;* ssdir: set dirbuf with current*
                          14384   ;*         ss pointer.           *
                          14385   ;*   regs: in: .a= low byte      *
                          14386   ;*********************************
                          14387
     DEDC                 14388   ssdir
     DEDC   85 94         14389           sta  dirbuf
     DEDE   A6 82         14390           ldx  lindx
     DEE0   B5 CD         14391           lda  ss,x
     DEE2   AA            14392           tax
     DEE3   BD FEE0       14393           lda  bufind,x
     DEE6   85 95         14394           sta  dirbuf+1
     DEE8   60            14395           rts
                          14396
                          14397   ;*********************************
                          14398   ;* setssp: set dirbuf & buftab   *
                          14399   ;*         with current ss ptr.  *
                          14400   ;*   regs: in: .a= low byte      *
                          14401   ;*********************************
                          14402
     DEE9                 14403   setssp
     DEE9   48            14404           pha
     DEEA   20 DEDC       14405           jsr  ssdir
     DEED   48            14406           pha
     DEEE   8A            14407           txa
     DEEF   0A            14408           asl  a
     DEF0   AA            14409           tax
     DEF1   68            14410           pla
     DEF2   95 9A         14411           sta  buftab+1,x
     DEF4   68            14412           pla
     DEF5   95 99         14413           sta  buftab,x
     DEF7   60            14414           rts
                          14415
                          14416   ;*********************************
                          14417   ;* sspos: position ss & buftab   *
                          14418   ;*         to ssnum ssind.       *
                          14419   ;*   flag:   .v=0: ok            *
                          14420   ;*           .v=1: out of range  *
                          14421   ;*********************************
```

```
Error Addr  Code           Seq    Source statement

                          14422
     DEF8                 14423  sspos
     DEF8   20 DF66       14424          jsr  sstest
     DEFB   30 0E         14425          bmi  ssp10      ; out of range
     DEFD   50 13         14426          bvc  ssp20      ; er0:ok, in range
                          14427
     DEFF   A6 82         14428          ldx  lindx      ; er1: possibly in range
     DF01   B5 CD         14429          lda  ss,x
     DF03   20 DF1B       14430          jsr  ibrd       ; read ss in
     DF06   20 DF66       14431          jsr  sstest     ; test again
     DF09   10 07         14432          bpl  ssp20
     DF0B                 14433  ssp10
     DF0B   20 E1CB       14434          jsr  ssend      ; not in range,set end
     DF0E   2C FECE       14435          bit  er1
     DF11   60            14436          rts
     DF12                 14437  ssp20
     DF12   A5 D6         14438          lda  ssind      ; ok, set ptr w/ index
     DF14   20 DEE9       14439          jsr  setssp
     DF17   2C FECD       14440          bit  er0
     DF1A   60            14441          rts
                          14442
                          14443  ;*********************************
                          14444  ;* ibrd: indirect block read &   *
                          14445  ;* ibwt: write.                  *
                          14446  ;*   regs: in: .a= buf # for r/w *
                          14447  ;*             .x= lindx         *
                          14448  ;*         (dirbuf),y   points to *
                          14449  ;*         t&s to be r/w.        *
                          14450  ;*********************************
                          14451
     DF1B                 14452  ibrd
     DF1B   85 F9         14453          sta  jobnum
     DF1D   A9 80         14454          lda  #read
     DF1F   D0 04         14455          bne  ibop
     DF21                 14456  ibwt
     DF21   85 F9         14457          sta  jobnum
     DF23   A9 90         14458          lda  #write
     DF25                 14459  ibop
     DF25   48            14460          pha
     DF26   B5 EC         14461          lda  filtyp,x
     DF28   29 01         14462          and  #1
     DF2A   85 7F         14463          sta  drvnum
     DF2C   68            14464          pla
     DF2D   05 7F         14465          ora  drvnum
     DF2F   8D 024D       14466          sta  cmd
                          14467
     DF32   B1 94         14468          lda  (dirbuf),y
     DF34   85 80         14469          sta  track
     DF36   C8            14470          iny
     DF37   B1 94         14471          lda  (dirbuf),y
```

```
Error Addr  Code           Seq    Source statement

     DF39   85 81        14472            sta  sector
     DF3B   A5 F9        14473            lda  jobnum
     DF3D   20 D6D3      14474            jsr  seth
     DF40   A6 F9        14475            ldx  jobnum
     DF42   4C D593      14476            jmp  doit2
                         14477
                         14478
                         14479    ;*
                         14480    ;*****************************
                         14481    ;*
                         14482    ;*  gsspnt
                         14483    ;*
                         14484    ;*****************************
                         14485    ;*
     DF45   A6 82        14486    gsspnt  ldx  lindx
     DF47   B5 CD        14487            lda  ss,x
     DF49   4C D4EB      14488            jmp  gp1
                         14489
     DF4C                14490    scal1
     DF4C   A9 78        14491            lda  #nssp
     DF4E   20 DF5C      14492            jsr  addt12     ; add (#ss needed)*120
     DF51                14493    sscalc
     DF51   CA           14494            dex
     DF52   10 F8        14495            bpl  scal1
                         14496
     DF54   A5 72        14497            lda  t3         ; add (# ss indices needed)
     DF56   4A           14498            lsr  a
     DF57   20 DF5C      14499            jsr  addt12
     DF5A   A5 73        14500            lda  t4         ; add (# ss blocks needed)
                         14501
     DF5C                14502    addt12
     DF5C   18           14503            clc             ; add .a to t1,t2
     DF5D   65 70        14504            adc  t1
     DF5F   85 70        14505            sta  t1
     DF61   90 02        14506            bcc  addrts
     DF63   E6 71        14507            inc  t2
     DF65                14508    addrts
     DF65   60           14509            rts
                         14510            .include  sstest
                         14511    ;*********************************
                         14512    ;* sstst: test ssnum & ssind for *
                         14513    ;*        residence & range.     *
                         14514    ;*   vars:  ssnum,ssind,dirbuf   *
                         14515    ;*                               *
                         14516    ;*   flags: .n rng  .v res  er   *
                         14517    ;*           0  ok   0 yes  er0  *
                         14518    ;*           0 maybe 1 no   er1  *
                         14519    ;*           1 no    0 yes  er2  *
                         14520    ;*           1 no    1 no   er3  *
                         14521    ;*********************************
```

```
Error Addr  Code           Seq    Source statement

                          14522   ;
     DF66                 14523   sstest
     DF66   20 DED2       14524           jsr  ssset       ; set pointer & ss #
     DF69   C5 D5         14525           cmp  ssnum
     DF6B   D0 0E         14526           bne  st20        ; not this ss
                          14527   ;
     DF6D   A4 D6         14528           ldy  ssind
     DF6F   B1 94         14529           lda  (dirbuf),y
     DF71   F0 04         14530           beq  st10
     DF73   2C FECD       14531           bit  er0         ; ok, resident
     DF76   60            14532           rts
     DF77                 14533   st10
     DF77   2C FECF       14534           bit  er2         ; out of range, res
     DF7A   60            14535           rts
     DF7B                 14536   st20
     DF7B   A5 D5         14537           lda  ssnum
     DF7D   C9 06         14538           cmp  #nssl
     DF7F   B0 0A         14539           bcs  st30
     DF81   0A            14540           asl  a
     DF82   A8            14541           tay
     DF83   A9 04         14542           lda  #4
     DF85   85 94         14543           sta  dirbuf
     DF87   B1 94         14544           lda  (dirbuf),y
     DF89   D0 04         14545           bne  st40
     DF8B                 14546   st30
     DF8B   2C FED0       14547           bit  er3         ; way out of range
     DF8E   60            14548           rts
     DF8F                 14549   st40
     DF8F   2C FECE       14550           bit  er1         ; not res, range???
     DF92   60            14551           rts
                          14552   ;.end
                          14553           .include  getact
```

```
Error Addr  Code           Seq   Source statement

                          14555           .subttl 'getact.src'
                          14556   ;*********************************
                          14557   ;* getact: get active buffer #    *
                          14558   ;*    vars: buf0,buf1,lindx       *
                          14559   ;*    regs: out: .a= act buffer # *
                          14560   ;*               .x= lindx        *
                          14561   ;*    flags:     .n=1: no act-buf *
                          14562   ;*********************************
                          14563
     DF93                 14564   getact
     DF93   A6 82         14565           ldx  lindx
     DF95   B5 A7         14566           lda  buf0,x
     DF97   10 02         14567           bpl  ga1
     DF99   B5 AE         14568           lda  buf1,x
     DF9B                 14569   ga1
     DF9B   29 BF         14570           and  #$bf       ;  strip dirty bit
     DF9D   60            14571           rts
                          14572
                          14573   ;*********************************
                          14574   ;* gaflg: get active buffer #;    *
                          14575   ;*         set lbused & flags.    *
                          14576   ;*    regs: out: .a= act buffer # *
                          14577   ;*               .x= lindx        *
                          14578   ;*    flags:     .n=1: no act-buf *
                          14579   ;*               .v=1: dirty buf  *
                          14580   ;*********************************
                          14581
     DF9E                 14582   gaflgs
     DF9E   A6 82         14583           ldx  lindx
     DFA0   8E 0257       14584   ga2     stx  lbused     ; save buf #
     DFA3   B5 A7         14585           lda  buf0,x
     DFA5   10 09         14586           bpl  ga3
                          14587
     DFA7   8A            14588           txa
     DFA8   18            14589           clc
     DFA9   69 07         14590           adc  #mxchns+1
     DFAB   8D 0257       14591           sta  lbused
     DFAE   B5 AE         14592           lda  buf1,x
     DFB0                 14593   ga3
     DFB0   85 70         14594           sta  t1
     DFB2   29 1F         14595           and  #$1f
     DFB4   24 70         14596           bit  t1
     DFB6   60            14597           rts
                          14598
                          14599   ;*****************************
                          14600
                          14601   ; get channels inactive
                          14602   ; buffer number.
                          14603
                          14604   ;     input parameters:
```

```
Error Addr  Code           Seq    Source statement

                          14605   ;         lindx - channel #
                          14606
                          14607   ;     output parameters:
                          14608   ;         a <== inactive buffer #
                          14609   ;            or
                          14610   ;         a <== $ff if no
                          14611   ;             inactive buffer.
                          14612
                          14613   ;*****************************
                          14614
     DFB7   A6 82         14615   getina  ldx  lindx
     DFB9   B5 A7         14616           lda  buf0,x
     DFBB   30 02         14617           bmi  gi10
     DFBD   B5 AE         14618           lda  buf1,x
     DFBF   C9 FF         14619   gi10    cmp  #$ff
     DFC1   60            14620           rts
                          14621
                          14622   ;*****************************
                          14623   ;**********  p u t  i n  a  ****
                          14624   ;*****************************
                          14625
                          14626   ; put inactive buffer
                          14627
                          14628   ;     input paramters:
                          14629   ;         a = buffer #
                          14630
                          14631   ;     output paramters:
                          14632   ;         none
                          14633
                          14634   ;*****************************
                          14635
     DFC2   A6 82         14636   putina  ldx  lindx
     DFC4   09 80         14637           ora  #$80
     DFC6   B4 A7         14638           ldy  buf0,x
     DFC8   10 03         14639           bpl  pi1
     DFCA   95 A7         14640           sta  buf0,x
     DFCC   60            14641           rts
     DFCD   95 AE         14642   pi1     sta  buf1,x
     DFCF   60            14643           rts
                          14644           .include  rel1
```

```
Error Addr  Code           Seq    Source statement

                          14646            .subttl 'rel1.src'
                          14647   ;**********************************
                          14648   ;*
                          14649   ;*      routine: nxtrec
                          14650   ;*
                          14651   ;*
                          14652   ;*
                          14653   ;*
                          14654   ;*
                          14655   ;**********************************
     DFDO                 14656   nxtrec
     DFDO    A9 2O        14657           lda  #ovrflo
     DFD2    2O DD9D      14658           jsr  clrflg
                          14659
     DFD5    A9 8O        14660           lda  #lrf
     DFD7    2O DDA6      14661           jsr  tstflg
     DFDA    DO 41        14662           bne  nxtr4O
                          14663
     DFDC    A6 82        14664           ldx  lindx
     DFDE    F6 B5        14665           inc  recl,x      ;  goto next record #
     DFEO    DO O2        14666           bne  nxtr15
     DFE2    F6 BB        14667           inc  rech,x
                          14668
     DFE4                 14669   nxtr15
     DFE4    A6 82        14670           ldx  lindx
     DFE6    B5 C1        14671           lda  nr,x
     DFE8    FO 2E        14672           beq  nxtr45      ; there is a nr
                          14673
     DFEA    2O D4E8      14674           jsr  getpnt      ;  get pointer
     DFED    A6 82        14675           ldx  lindx       ; test if same buffer
     DFEF    D5 C1        14676           cmp  nr,x
     DFF1    9O O3        14677           bcc  nxtr2O      ;  yes, bt<nr
                          14678
     DFF3    2O EO3C      14679           jsr  nrbuf       ;  no,next buffer
                          14680
     DFF6    A6 82        14681   nxtr2O  ldx  lindx
     DFF8    B5 C1        14682           lda  nr,x
     DFFA    2O D4C8      14683           jsr  setpnt      ; advance to next rec
     DFFD    A1 99        14684           lda  (buftab,x)  ; read 1st dat byte
                          14685
     DFFF    85 85        14686           sta  data        ; save for read channel
     EOO1    A9 2O        14687           lda  #ovrflo
     EOO3    2O DD9D      14688           jsr  clrflg      ;  clear
                          14689   ; the overflow flag
     EOO6    2O E3O4      14690           jsr  addnr       ; advance nr
     EOO9                 14691   nxout
     EOO9    48           14692           pha
     EOOA    9O 28        14693           bcc  nxtr3O      ; no block boundary
                          14694
     EOOC    A9 OO        14695           lda  #O
```

```
Error Addr  Code           Seq    Source statement

     E00E   20 D4F6      14696            jsr  drdbyt      ; check track link
     E011   D0 21        14697            bne  nxtr30      ; not last block
                         14698
     E013   68           14699            pla
     E014   C9 02        14700            cmp  #2
     E016   F0 12        14701            beq  nxtr50
     E018                14702   nxtr45
     E018   A9 80        14703            lda  #lrf
     E01A   20 DD97      14704            jsr  setflg
     E01D                14705   nxtr40
     E01D   20 D12F      14706            jsr  getpre
     E020   B5 99        14707            lda  buftab,x
     E022   99 0244      14708            sta  lstchr,y
     E025   A9 0D        14709            lda  #cr
     E027   85 85        14710            sta  data
     E029   60           14711            rts
                         14712
     E02A                14713   nxtr50
     E02A   20 E035      14714            jsr  nxtr35
     E02D   A6 82        14715            ldx  lindx
     E02F   A9 00        14716            lda  #0
     E031   95 C1        14717            sta  nr,x
     E033   60           14718            rts
     E034                14719   nxtr30
     E034   68           14720            pla
     E035                14721   nxtr35
     E035   A6 82        14722            ldx  lindx
                         14723   ;*
     E037   95 C1        14724            sta  nr,x
     E039   4C E16E      14725            jmp  setlst
                         14726   ;*
                         14727   ;*
                         14728   ;*********************************
                         14729   ;*
                         14730   ;*
                         14731   ;*  nrbuf
                         14732   ;*
                         14733   ;*
                         14734   ;*********************************
                         14735   ;*
                         14736   ;*
     E03C                14737   nrbuf                        ; read trk,sec link
     E03C   20 D1D3      14738            jsr  setdrn
     E03F   20 DE95      14739            jsr  rdlnk
                         14740
     E042   20 DF9E      14741            jsr  gaflgs      ; test if dirty
     E045   50 16        14742            bvc  nrbu50      ; clean, dont write out
                         14743
     E047   20 DE5E      14744            jsr  wrtout      ; dirty, write out
     E04A   20 CF1E      14745            jsr  dblbuf      ; toggle active buffer
```

```
Error Addr  Code           Seq    Source statement

                          14746
      E04D  A9 02         14747            lda  #2
      E04F  20 D4C8       14748            jsr  setpnt
      E052  20 DDAB       14749            jsr  tstwrt      ; test if lstjob is wrt
      E055  D0 24         14750            bne  nrbu20      ; not a write,buffer ok
                          14751
      E057  20 DE57       14752            jsr  rdab        ; read in needed buffer
      E05A  4C D599       14753            jmp  watjob      ; wait around till done
                          14754
      E05D  20 CF1E       14755  nrbu50    jsr  dblbuf      ; toggle act buf
      E060  20 DDAB       14756            jsr  tstwrt      ; was lstjob a wrt?
      E063  D0 06         14757            bne  nrbu70      ; not a write
                          14758
      E065  20 DE57       14759            jsr  rdab        ; read in needed buffer
      E068  20 D599       14760            jsr  watjob      ; wait till done
                          14761
      E06B                14762  nrbu70                     ; read trk,sec link
      E06B  20 DE95       14763            jsr  rdlnk       ; to do a read ahead
                          14764
      E06E  A5 80         14765            lda  track       ; test if last buffer
      E070  F0 09         14766            beq  nrbu20      ; yes,no dbl buff todo
                          14767
      E072  20 CF1E       14768            jsr  dblbuf      ; start read job on the
      E075  20 DE57       14769            jsr  rdab        ; inactive buffer
      E078  20 CF1E       14770            jsr  dblbuf
                          14771
      E07B  60            14772  nrbu20    rts
                          14773            .include  rel2
```

```
Error Addr  Code           Seq    Source statement

                          14775           .subttl 'rel2.src'
                          14776   ;*********************************
                          14777   ;*                                 .
                          14778   ;* relput
                          14779   ;*
                          14780   ;*
                          14781   ;*********************************
                          14782   ;*
                          14783   ;*
     E07C   20 E105       14784   relput  jsr  sdirty     ; write data to buffer
     E07F   20 DF93       14785           jsr  getact
     E082   0A            14786           asl  a
     E083   AA            14787           tax
     E084   A5 85         14788           lda  data
     E086   81 99         14789           sta  (buftab,x)
                          14790
     E088   B4 99         14791           ldy  buftab,x   ; inc the pointer
     E08A   C8            14792           iny
     E08B   D0 09         14793           bne  relp05
     E08D   A4 82         14794           ldy  lindx
     E08F   B9 00C1       14795           lda  nr,y
     E092   F0 0A         14796           beq  relp07
                          14797
     E094                 14798   relp06
     E094   A0 02         14799           ldy  #2
     E096                 14800   relp05
     E096   98            14801           tya
     E097   A4 82         14802           ldy  lindx
                          14803
     E099   D9 00C1       14804           cmp  nr,y       ; test if nr=pointer
     E09C   D0 05         14805           bne  relp10     ; no,set new pointer
                          14806
     E09E   A9 20         14807   relp07  lda  #ovrflo    ; yes,set overflow
     E0A0   4C DD97       14808           jmp  setflg
                          14809
     E0A3                 14810   relp10                  ; write back new pointer
     E0A3   F6 99         14811           inc  buftab,x
                          14812
     E0A5   D0 03         14813           bne  relp20     ; test if =0
     E0A7   20 E03C       14814           jsr  nrbuf      ; prepare nxt buffer
                          14815
     E0AA   60            14816   relp20  rts
                          14817   ;*
                          14818   ;*
                          14819   ;*
                          14820   ;*********************************
                          14821   ;*
                          14822   ;*  wrtrel
                          14823   ;*
                          14824   ;*
```

```
Error Addr  Code           Seq    Source statement

                          14825   ;*********************************
                          14826   ;*
                          14827   ;*
     EOAB                 14828   wrtrel
     EOAB   A9 AO         14829           lda  #lrf+ovrflo ;check all flags
     EOAD   20 DDA6       14830           jsr  tstflg
     EOBO   DO 27         14831           bne  wr50       ; some flag is set
     EOB2                 14832   wr10
     EOB2   A5 85         14833           lda  data       ; ready to put data
     EOB4   20 EO7C       14834           jsr  relput
     EOB7                 14835   wr20
     EOB7   A5 F8         14836           lda  eoiflg
     EOB9   FO OD         14837           beq  wr40       ; eoi was sent
     EOBB   60            14838           rts
     EOBC                 14839   wr30
     EOBC   A9 20         14840           lda  #ovrflo
     EOBE   20 DDA6       14841           jsr  tstflg
     EOC1   FO O5         14842           beq  wr40       ; no rec overflow
     EOC3   A9 51         14843           lda  #recovf
     EOC5   8D O26C       14844           sta  erword     ; set error for end of print
     EOC8                 14845   wr40
     EOC8   20 EOF3       14846           jsr  clrec      ; clear rest of record
     EOCB   20 E153       14847           jsr  rd40
     EOCE   AD O26C       14848           lda  erword
     EOD1   FO O3         14849           beq  wr45
     EOD3   4C C1C8       14850           jmp  cmderr
     EOD6                 14851   wr45
     EOD6   4C E6BC       14852           jmp  okerr
                          14853
     EOD9                 14854   wr50
     EOD9   29 80         14855           and  #lrf
     EODB   DO O5         14856           bne  wr60       ; last rec, add
     EODD   A5 F8         14857           lda  eoiflg
     EODF   FO DB         14858           beq  wr30
     EOE1                 14859   wr51
     EOE1   60            14860           rts
     EOE2                 14861   wr60
     EOE2   A5 85         14862           lda  data
     EOE4   48            14863           pha
     EOE5   20 E31C       14864           jsr  addrel     ; add to file
     EOE8   68            14865           pla
     EOE9   85 85         14866           sta  data
     EOEB   A9 80         14867           lda  #lrf
     EOED   20 DD9D       14868           jsr  clrflg
     EOFO   4C EOB2       14869           jmp  wr10
                          14870   ;*
                          14871   ;*
                          14872   ;*
                          14873   ;*********************************
                          14874   ;*
```

```
Error Addr  Code           Seq    Source statement

                          14875   ;*   clrec
                          14876   ;*
                          14877   ;*********************************
                          14878
     E0F3   A9 20         14879   clrec   lda  #ovrflo    ; put 0's into rest of record
     E0F5   20 DDA6       14880           jsr  tstflg
     E0F8   D0 0A         14881           bne  clr10
                          14882
     E0FA   A9 00         14883           lda  #0
     E0FC   85 85         14884           sta  data
     E0FE   20 E07C       14885           jsr  relput
                          14886
     E101   4C E0F3       14887           jmp  clrec
                          14888
     E104   60            14889   clr10   rts
                          14890
                          14891   ;*
                          14892   ;*
                          14893   ;*********************************
                          14894   ;*
                          14895   ;*   sdirty
                          14896   ;*
                          14897   ;*********************************
                          14898   ;*
                          14899
     E105   A9 40         14900   sdirty  lda  #dyfile
     E107   20 DD97       14901           jsr  setflg
     E10A   20 DF9E       14902           jsr  gaflgs
     E10D   09 40         14903           ora  #$40
     E10F   AE 0257       14904           ldx  lbused
     E112   95 A7         14905           sta  buf0,x
     E114   60            14906           rts
                          14907
                          14908   ;*
                          14909   ;*
                          14910   ;*********************************
                          14911   ;*
                          14912   ;*   cdirty
                          14913   ;*
                          14914   ;*********************************
                          14915   ;*
                          14916
     E115   20 DF9E       14917   cdirty  jsr  gaflgs
     E118   29 BF         14918           and  #$bf
     E11A   AE 0257       14919           ldx  lbused
     E11D   95 A7         14920           sta  buf0,x
     E11F   60            14921           rts
                          14922           .include  rel3
```

```
Error Addr  Code           Seq    Source statement

                          14924            .subttl 'rel3.src'
                          14925   ;*********************************
                          14926   ;*
                          14927   ;*
                          14928   ;* rdrel
                          14929   ;*
                          14930   ;*
                          14931   ;*********************************
                          14932   ;*
                          14933   ;*
                          14934
     E120                 14935   rdrel
     E120   A9 80         14936            lda  #lrf
     E122   20 DDA6       14937            jsr  tstflg
     E125   D0 37         14938            bne  rd05       ; no record error
                          14939   ;
     E127                 14940   rd10
     E127   20 D12F       14941            jsr  getpre
     E12A   B5 99         14942            lda  buftab,x
     E12C   D9 0244       14943            cmp  lstchr,y
     E12F   F0 22         14944            beq  rd40
                          14945
     E131   F6 99         14946            inc  buftab,x
     E133   D0 06         14947            bne  rd20
                          14948
     E135   20 E03C       14949            jsr  nrbuf
     E138                 14950   rd15
     E138   20 D12F       14951            jsr  getpre
     E13B                 14952   rd20
     E13B   A1 99         14953            lda  (buftab,x)
     E13D                 14954   rd25
     E13D   99 023E       14955            sta  chndat,y
     E140   A9 89         14956            lda  #rndrdy
     E142   99 00F2       14957            sta  chnrdy,y
     E145   B5 99         14958            lda  buftab,x
     E147   D9 0244       14959            cmp  lstchr,y
     E14A   F0 01         14960            beq  rd30
     E14C   60            14961            rts
     E14D                 14962   rd30
     E14D   A9 81         14963            lda  #rndeoi
     E14F   99 00F2       14964            sta  chnrdy,y
     E152   60            14965            rts
     E153                 14966   rd40
     E153   20 DFD0       14967            jsr  nxtrec
     E156   20 D12F       14968            jsr  getpre
     E159   A5 85         14969            lda  data
     E15B   4C E13D       14970            jmp  rd25
                          14971
     E15E                 14972   rd05
     E15E   A6 82         14973            ldx  lindx      ; no record char set up
```

```
Error Addr  Code           Seq    Source statement

     E160   A9 0D        14974            lda  #cr
     E162   9D 023E      14975            sta  chndat,x
     E165   A9 81        14976            lda  #rndeoi
     E167   95 F2        14977            sta  chnrdy,x
                         14978    ;no record error
     E169   A9 50        14979            lda  #norec
     E16B   20 C1C8      14980            jsr  cmderr
                         14981            .include  rel4
```

```
Error Addr  Code           Seq    Source statement

                          14983            .subttl 'rel4.src'
                          14984    ;*
                          14985    ;*
                          14986    ;**********************************
                          14987    ;*
                          14988    ;*
                          14989    ;*  setlst
                          14990    ;*
                          14991    ;*
                          14992    ;**********************************
                          14993    ;*
                          14994    ;*
     E16E   A6 82         14995    setlst  ldx  lindx
     E170   B5 C1         14996            lda  nr,x
     E172   85 87         14997            sta  r1
     E174   C6 87         14998            dec  r1
     E176   C9 02         14999            cmp  #2
     E178   D0 04         15000            bne  setl01
     E17A   A9 FF         15001            lda  #$ff
     E17C   85 87         15002            sta  r1
     E17E                 15003    setl01
     E17E   B5 C7         15004            lda  rs,x
     E180   85 88         15005            sta  r2
                          15006
     E182   20 D4E8       15007            jsr  getpnt
     E185   A6 82         15008            ldx  lindx
     E187   C5 87         15009            cmp  r1
     E189   90 19         15010            bcc  setl10
     E18B   F0 17         15011            beq  setl10
                          15012
     E18D   20 CF1E       15013            jsr  dblbuf
     E190   20 E1B2       15014            jsr  fndlst
     E193   90 08         15015            bcc  setl05
                          15016
     E195   A6 82         15017            ldx  lindx
     E197   9D 0244       15018            sta  lstchr,x
     E19A   4C CF1E       15019            jmp  dblbuf
                          15020
     E19D   20 CF1E       15021    setl05  jsr  dblbuf
     E1A0   A9 FF         15022            lda  #$ff
     E1A2   85 87         15023            sta  r1
                          15024
     E1A4   20 E1B2       15025    setl10  jsr  fndlst
     E1A7   B0 03         15026            bcs  setl40
                          15027
     E1A9   20 D4E8       15028            jsr  getpnt
                          15029
     E1AC   A6 82         15030    setl40  ldx  lindx
     E1AE   9D 0244       15031            sta  lstchr,x
     E1B1   60            15032            rts
```

```
Error Addr  Code           Seq    Source statement

                          15033   ;*
                          15034   ;*
                          15035   ;*
                          15036   ;*********************************
                          15037   ;*
                          15038   ;*
                          15039   ;*  fndlst
                          15040   ;*
                          15041   ;*
                          15042   ;*********************************
                          15043   ;*
                          15044   ;*
     E1B2                 15045   fndlst
     E1B2   20 DE2B       15046           jsr  set00
     E1B5   A4 87         15047           ldy  r1          ; offset to start at
                          15048
     E1B7   B1 94         15049   fndl10  lda  (dirbuf),y
     E1B9   D0 0D         15050           bne  fndl20
                          15051
     E1BB   88            15052           dey
     E1BC   C0 02         15053           cpy  #2
     E1BE   90 04         15054           bcc  fndl30
                          15055
     E1C0   C6 88         15056           dec  r2          ; limit counter
     E1C2   D0 F3         15057           bne  fndl10
                          15058
     E1C4   C6 88         15059   fndl30  dec  r2
     E1C6   18            15060           clc              ;  not found here
     E1C7   60            15061           rts
                          15062
     E1C8   98            15063   fndl20  tya              ; found the end char
     E1C9   38            15064           sec
     E1CA   60            15065           rts
                          15066           .include  ssend
```

```
Error Addr  Code           Seq    Source statement

                          15068           .subttl 'ssend.src'
                          15069   ;*********************************
                          15070   ;* ssend: position ss & buftab   *
                          15071   ;*         to end of last record. *
                          15072   ;*   vars:                        *
                          15073   ;*********************************
                          15074
     E1CB                 15075   ssend
     E1CB   20 DED2       15076           jsr  ssset
     E1CE   85 D5         15077           sta  ssnum
     E1D0   A9 04         15078           lda  #4
     E1D2   85 94         15079           sta  dirbuf
     E1D4   A0 0A         15080           ldy  #ssioff-6
     E1D6   D0 04         15081           bne  se20
     E1D8                 15082   se10
     E1D8   88            15083           dey
     E1D9   88            15084           dey
     E1DA   30 26         15085           bmi  break
     E1DC                 15086   se20
     E1DC   B1 94         15087           lda  (dirbuf),y ; look for last ss #
     E1DE   F0 F8         15088           beq  se10       ; t=0: not yet
     E1E0   98            15089           tya
     E1E1   4A            15090           lsr  a
     E1E2   C5 D5         15091           cmp  ssnum      ; check ss #
     E1E4   F0 09         15092           beq  se30       ; this is last ss
                          15093
     E1E6   85 D5         15094           sta  ssnum
     E1E8   A6 82         15095           ldx  lindx
     E1EA   B5 CD         15096           lda  ss,x
     E1EC   20 DF1B       15097           jsr  ibrd       ; read last ss
     E1EF                 15098   se30
     E1EF   A0 00         15099           ldy  #0         ; set ssind
     E1F1   84 94         15100           sty  dirbuf
     E1F3   B1 94         15101           lda  (dirbuf),y ; debug
     E1F5   D0 0B         15102           bne  break
     E1F7   C8            15103           iny
     E1F8   B1 94         15104           lda  (dirbuf),y
     E1FA   A8            15105           tay             ; back up to track
     E1FB   88            15106           dey
     E1FC   84 D6         15107           sty  ssind
     E1FE   98            15108           tya
     E1FF   4C DEE9       15109           jmp  setssp
                          15110
     E202                 15111   break
     E202   A9 67         15112           lda  #$67
     E204   20 E645       15113           jsr  cmder2
                          15114           .include  record
```

```
Error Addr  Code           Seq    Source statement

                          15116           .subttl 'record.src'
                          15117   ;*********************************
                          15118   ;* record: position relative     *
                          15119   ;*         pointers to given     *
                          15120   ;*         record number or to   *
                          15121   ;*         last record if out of *
                          15122   ;*         range.                *
                          15123   ;*********************************
     E207                 15124   record
     E207   20 C2B3       15125           jsr  cmdset     ; init tables, ptrs
     E20A   AD 0201       15126           lda  cmdbuf+1
     E20D   85 83         15127           sta  sa
     E20F   20 D0EB       15128           jsr  fndrch
     E212   90 05         15129           bcc  r20        ; got channel's lindex
                          15130
     E214   A9 70         15131           lda  #nochnl    ; no valid channel
     E216   20 C1C8       15132           jsr  cmderr
     E219                 15133   r20
     E219   A9 A0         15134           lda  #lrf+ovrflo
     E21B   20 DD9D       15135           jsr  clrflg
     E21E   20 D125       15136           jsr  typfil     ; get file type
     E221   F0 05         15137           beq  r30        ; it is relative file
                          15138
     E223   A9 64         15139           lda  #mistyp    ; wrong type
     E225   20 C1C8       15140           jsr  cmderr
     E228                 15141   r30
     E228   B5 EC         15142           lda  filtyp,x
     E22A   29 01         15143           and  #1
     E22C   85 7F         15144           sta  drvnum
     E22E   AD 0202       15145           lda  cmdbuf+2
     E231   95 B5         15146           sta  recl,x     ; get record #
     E233   AD 0203       15147           lda  cmdbuf+3
     E236   95 BB         15148           sta  rech,x
     E238   A6 82         15149           ldx  lindx      ; clear chnrdy to rndrdy
     E23A   A9 89         15150           lda  #rndrdy
     E23C   95 F2         15151           sta  chnrdy,x
                          15152
     E23E   AD 0204       15153           lda  cmdbuf+4   ; get offset
     E241   F0 10         15154           beq  r40
     E243   38            15155           sec
     E244   E9 01         15156           sbc  #1
     E246   F0 0B         15157           beq  r40
     E248   D5 C7         15158           cmp  rs,x
     E24A   90 07         15159           bcc  r35
                          15160
     E24C   A9 51         15161           lda  #recovf
     E24E   8D 026C       15162           sta  erword
     E251   A9 00         15163           lda  #0
     E253                 15164   r35
     E253                 15165   r40
```

```
Error Addr  Code          Seq   Source statement

     E253   85 D4       15166           sta  recptr      ; set offset
     E255   20 CE0E     15167           jsr  fndrel      ; calc ss stuff
     E258   20 DEF8     15168           jsr  sspos       ; set ss ptrs
     E25B   50 08       15169           bvc  r50
                        15170
     E25D   A9 80       15171           lda  #lrf        ; beyond the end
     E25F   20 DD97     15172           jsr  setflg      ; set last rec flag
     E262   4C E15E     15173           jmp  rd05
     E265               15174   r50
     E265   20 E275     15175           jsr  positn      ; position to record
     E268   A9 80       15176           lda  #lrf
     E26A   20 DDA6     15177           jsr  tstflg
     E26D   F0 03       15178           beq  r60
     E26F   4C E15E     15179           jmp  rd05
     E272               15180   r60
     E272   4C C194     15181           jmp  endcmd      ; that's all
                        15182
                        15183   ;*********************************
                        15184   ;* positn: position relative     *
                        15185   ;*         data block into active*
                        15186   ;*         buffer & next block   *
                        15187   ;*         into inactive buffer. *
                        15188   ;*********************************
                        15189
     E275               15190   positn
     E275   20 E29C     15191           jsr  posbuf      ; position buffers
     E278   A5 D7       15192           lda  relptr
     E27A   20 D4C8     15193           jsr  setpnt      ; set ptr from fndrel
                        15194
     E27D   A6 82       15195           ldx  lindx
     E27F   B5 C7       15196           lda  rs,x
     E281   38          15197           sec              ; calc the offset
     E282   E5 D4       15198           sbc  recptr
     E284   B0 03       15199           bcs  p2
     E286   4C E202     15200           jmp  break       ; should not be needed
     E289               15201   p2
     E289   18          15202           clc
     E28A   65 D7       15203           adc  relptr
     E28C   90 03       15204           bcc  p30
     E28E   69 01       15205           adc  #1
     E290   38          15206           sec
     E291               15207   p30
     E291   20 E009     15208           jsr  nxout       ; set nr
     E294   4C E138     15209           jmp  rd15
     E297   A9 51       15210           lda  #recovf
     E299   20 C1C8     15211           jsr  cmderr
                        15212   ;*
                        15213   ;*********************************
                        15214   ;* posbuf: position proper data  *
                        15215   ;*         blocks into buffers   *
```

```
Error Addr  Code           Seq    Source statement

                          15216   ;*********************************
                          15217
     E29C                 15218   posbuf
     E29C   A5 94         15219           lda  dirbuf
     E29E   85 89         15220           sta  r3
     E2A0   A5 95         15221           lda  dirbuf+1
     E2A2   85 8A         15222           sta  r4
     E2A4   20 E2D0       15223           jsr  bhere      ; is buffer in?
     E2A7   D0 01         15224           bne  p10        ; yes!
     E2A9   60            15225           rts
     E2AA                 15226   p10
     E2AA   20 DDF1       15227           jsr  scrub      ; clean buffer
     E2AD   20 DE0C       15228           jsr  getlnk
     E2B0   A5 80         15229           lda  track
     E2B2   F0 0E         15230           beq  p80
     E2B4   20 E2D3       15231           jsr  bhere2
     E2B7   D0 06         15232           bne  p75
     E2B9   20 CF1E       15233           jsr  dblbuf
     E2BC   4C D2DA       15234           jmp  freiac
     E2BF                 15235   p75
     E2BF   20 D2DA       15236           jsr  freiac
                          15237
     E2C2                 15238   p80
     E2C2   A0 00         15239           ldy  #0         ; get proper block
     E2C4   B1 89         15240           lda  (r3),y
     E2C6   85 80         15241           sta  track
     E2C8   C8            15242           iny
     E2C9   B1 89         15243           lda  (r3),y
     E2CB   85 81         15244           sta  sector
     E2CD   4C D0AF       15245           jmp  strdbl     ; get next block, too.
     E2D0                 15246   bhere
     E2D0   20 DE3E       15247           jsr  gethdr     ; get the header
     E2D3                 15248   bhere2
     E2D3   A0 00         15249           ldy  #0
     E2D5   B1 89         15250           lda  (r3),y
     E2D7   C5 80         15251           cmp  track
     E2D9   F0 01         15252           beq  bh10       ; test sector, too.
     E2DB   60            15253           rts
     E2DC                 15254   bh10
     E2DC   C8            15255           iny
     E2DD   B1 89         15256           lda  (r3),y
     E2DF   C5 81         15257           cmp  sector     ; set .z
     E2E1   60            15258           rts
                          15259           .include  nulbuf
```

```
Error Addr  Code           Seq    Source statement

                          15261           .subttl 'nulbuf.src'
                          15262   ;*********************************
                          15263   ;* nulbuf: set null records in   *
                          15264   ;*          act-buf for extention *
                          15265   ;*     vars:  nr,rs,lx,act-buf    *
                          15266   ;*       in: nr= last record      *
                          15267   ;*           position in previous *
                          15268   ;*           buffer.              *
                          15269   ;*      out: nr= last record      *
                          15270   ;*           position in buffer   *
                          15271   ;*           for next nulbuf or   *
                          15272   ;*           to set lstchr.       *
                          15273   ;*********************************
                          15274
     E2E2                 15275   nulbuf
     E2E2   20 DE2B       15276           jsr  set00      ; set indirect ptr
     E2E5   A0 02         15277           ldy  #2
     E2E7   A9 00         15278           lda  #0
     E2E9                 15279   nb10
     E2E9   91 94         15280           sta  (dirbuf),y ; clear buffer
     E2EB   C8            15281           iny
     E2EC   D0 FB         15282           bne  nb10
                          15283
     E2EE   20 E304       15284           jsr  addnr      ; advance nr
     E2F1                 15285   nb20
     E2F1   95 C1         15286           sta  nr,x
     E2F3   A8            15287           tay
     E2F4   A9 FF         15288           lda  #$ff
     E2F6   91 94         15289           sta  (dirbuf),y ; init record w/ cr
     E2F8   20 E304       15290           jsr  addnr
     E2FB   90 F4         15291           bcc  nb20       ; not done
                          15292
     E2FD   D0 04         15293           bne  nb30
     E2FF   A9 00         15294           lda  #0
     E301   95 C1         15295           sta  nr,x
     E303                 15296   nb30
     E303   60            15297           rts
                          15298
                          15299   ; add rs & nr, leave in accum
                          15300   ; c=1: cross buffer boundary
                          15301
     E304                 15302   addnr
     E304   A6 82         15303           ldx  lindx
     E306   B5 C1         15304           lda  nr,x
     E308   38            15305           sec
     E309   F0 0D         15306           beq  an05
                          15307
     E30B   18            15308           clc
     E30C   75 C7         15309           adc  rs,x
     E30E   90 0B         15310           bcc  an10
```

```
Error Addr  Code           Seq    Source statement

     E310   D0 06        15311            bne  an05
     E312   A9 02        15312            lda  #2
     E314   2C FECC      15313            bit  er00
     E317   60           15314            rts
                         15315
     E318                15316    an05
     E318   69 01        15317            adc  #1          ; adjust for link
     E31A   38           15318            sec
     E31B                15319    an10
     E31B   60           15320            rts
                         15321            .include  addrel
```

```
Error Addr  Code           Seq    Source statement

                          15323           .subttl 'addrel.src'
                          15324   ;*********************************
                          15325   ;* addrel: add blocks to relative*
                          15326   ;*         file.                 *
                          15327   ;*   vars:                       *
                          15328   ;*   regs:                       *
                          15329   ;*                               *
                          15330   ;*********************************
                          15331
     E31C                 15332   addrel
     E31C   20 D1D3       15333           jsr  setdrn
     E31F   20 E1CB       15334           jsr  ssend      ; set up end of file
     E322   20 E29C       15335           jsr  posbuf
     E325   20 CF7B       15336           jsr  dbset
     E328   A5 D6         15337           lda  ssind
     E32A   85 87         15338           sta  r1         ; save ss index
     E32C   A5 D5         15339           lda  ssnum
     E32E   85 86         15340           sta  r0         ; save ss number
     E330   A9 00         15341           lda  #0
     E332   85 88         15342           sta  r2         ; clear flag for one block
                          15343
     E334   A9 00         15344           lda  #0         ; clear for calculation...
     E336   85 D4         15345           sta  recptr     ; ...to 1st byte in record
     E338   20 CE0E       15346           jsr  fndrel     ; calc ss ptrs
     E33B                 15347   addr1                   ;  entry for rel record fix
     E33B   20 EF4D       15348           jsr  numfre     ; calc available...
                          15349
     E33E   A4 82         15350           ldy  lindx      ; record span?
     E340   B6 C7         15351           ldx  rs,y
     E342   CA            15352           dex
     E343   8A            15353           txa
     E344   18            15354           clc
     E345   65 D7         15355           adc  relptr
     E347   90 0C         15356           bcc  ar10       ; no span
                          15357
     E349   E6 D6         15358           inc  ssind      ; inc ss ptrs & check
     E34B   E6 D6         15359           inc  ssind      ; inc ss ptrs & check
     E34D   D0 06         15360           bne  ar10
     E34F   E6 D5         15361           inc  ssnum
     E351   A9 10         15362           lda  #ssioff
     E353   85 D6         15363           sta  ssind
     E355                 15364   ar10
     E355   A5 87         15365           lda  r1
     E357   18            15366           clc
     E358   69 02         15367           adc  #2
     E35A   20 DEE9       15368           jsr  setssp
                          15369
     E35D   A5 D5         15370           lda  ssnum
     E35F   C9 06         15371           cmp  #nssl
     E361   90 05         15372           bcc  ar25       ; valid range
```

```
Error Addr  Code           Seq    Source statement

                          15373
      E363                15374   ar20
      E363   A9 52        15375           lda  #bigfil
      E365   20 C1C8      15376           jsr  cmderr     ; too many ss's
      E368                15377   ar25
      E368   A5 D6        15378           lda  ssind      ; calc # blocks needed...
      E36A   38           15379           sec             ; ...& check against avail.
      E36B   E5 87        15380           sbc  r1
      E36D   B0 03        15381           bcs  ar30
      E36F   E9 0F        15382           sbc  #ssioff-1
      E371   18           15383           clc
      E372                15384   ar30
      E372   85 72        15385           sta  t3         ; # ss indices
      E374   A5 D5        15386           lda  ssnum
      E376   E5 86        15387           sbc  r0
      E378   85 73        15388           sta  t4         ; # ss needed
                          15389
      E37A   A2 00        15390           ldx  #0         ; clear accum.
      E37C   86 70        15391           stx  t1
      E37E   86 71        15392           stx  t2
      E380   AA           15393           tax             ; .x=# ss
      E381   20 DF51      15394           jsr  sscalc     ; calc # of blocks needed
                          15395
      E384   A5 71        15396           lda  t2
      E386   D0 07        15397           bne  ar35
      E388   A6 70        15398           ldx  t1
      E38A   CA           15399           dex
      E38B   D0 02        15400           bne  ar35
                          15401
      E38D   E6 88        15402           inc  r2
      E38F                15403   ar35
      E38F   CD 0273      15404           cmp  nbtemp+1
      E392   90 09        15405           bcc  ar40       ; ok!!
      E394   D0 CD        15406           bne  ar20
      E396   AD 0272      15407           lda  nbtemp
      E399   C5 70        15408           cmp  t1
      E39B   90 C6        15409           bcc  ar20       ; not enuf blocks
      E39D                15410   ar40
      E39D   A9 01        15411           lda  #1
      E39F   20 D4F6      15412           jsr  drdbyt     ; look at sector link
      E3A2   18           15413           clc
      E3A3   69 01        15414           adc  #1         ; +1 is nr
      E3A5   A6 82        15415           ldx  lindx
      E3A7   95 C1        15416           sta  nr,x
      E3A9   20 F11E      15417           jsr  nxtts      ; get next block...
      E3AC   20 DDFD      15418           jsr  setlnk     ; ...& set link.
      E3AF   A5 88        15419           lda  r2
      E3B1   D0 15        15420           bne  ar50       ; add one block
                          15421
      E3B3   20 DE5E      15422           jsr  wrtout     ; write current last rec
```

```
Error Addr  Code           Seq    Source statement

     E3B6                15423   ar45
     E3B6   20 CF1E      15424           jsr  dblbuf      ; switch bufs
     E3B9   20 D6D0      15425           jsr  sethdr      ; set hdr from t & s
     E3BC   20 F11E      15426           jsr  nxtts       ; get another
     E3BF   20 DDFD      15427           jsr  setlnk      ; set up link
     E3C2   20 E2E2      15428           jsr  nulbuf      ; clean it out
     E3C5   4C E3D4      15429           jmp  ar55
     E3C8                15430   ar50
     E3C8   20 CF1E      15431           jsr  dblbuf      ; switch bufs
     E3CB   20 D6D0      15432           jsr  sethdr      ; set hdr from t & s
     E3CE   20 E2E2      15433           jsr  nulbuf      ; clean buffer
     E3D1   20 DE19      15434           jsr  nullnk      ; last block =0,lstchr
     E3D4                15435   ar55
     E3D4   20 DE5E      15436           jsr  wrtout      ; write buffer
     E3D7   20 DE0C      15437           jsr  getlnk      ; get t&s from link
     E3DA   A5 80        15438           lda  track
     E3DC   48           15439           pha              ; save 'em
     E3DD   A5 81        15440           lda  sector
     E3DF   48           15441           pha
     E3E0   20 DE3E      15442           jsr  gethdr      ; now get hdr t&s
     E3E3   A5 81        15443           lda  sector
     E3E5   48           15444           pha              ; save 'em
     E3E6   A5 80        15445           lda  track
     E3E8   48           15446           pha
     E3E9   20 DF45      15447           jsr  gsspnt      ; check ss ptr
     E3EC   AA           15448           tax
     E3ED   D0 0A        15449           bne  ar60
                         15450
     E3EF   20 E44E      15451           jsr  newss       ; need another ss
     E3F2   A9 10        15452           lda  #ssioff
     E3F4   20 DEE9      15453           jsr  setssp      ; .a=bt val
     E3F7   E6 86        15454           inc  r0          ; advance ss count
     E3F9                15455   ar60
     E3F9   68           15456           pla
     E3FA   20 DD8D      15457           jsr  putss       ; record t&s...
     E3FD   68           15458           pla
     E3FE   20 DD8D      15459           jsr  putss       ; ...in ss.
     E401   68           15460           pla              ; get t&s from link
     E402   85 81        15461           sta  sector
     E404   68           15462           pla
     E405   85 80        15463           sta  track
     E407   F0 0F        15464           beq  ar65        ; t=0: that's all!!
                         15465
     E409   A5 86        15466           lda  r0
     E40B   C5 D5        15467           cmp  ssnum
     E40D   D0 A7        15468           bne  ar45        ; not even done yet
                         15469
     E40F   20 DF45      15470           jsr  gsspnt
     E412   C5 D6        15471           cmp  ssind
     E414   90 A0        15472           bcc  ar45        ; almost done
```

```
Error Addr  Code           Seq    Source statement

     E416   F0 B0        15473            beq  ar50       ; one more block left
     E418                15474   ar65
     E418   20 DF45      15475            jsr  gsspnt
     E41B   48           15476            pha
     E41C   A9 00        15477            lda  #0
     E41E   20 DEDC      15478            jsr  ssdir
     E421   A9 00        15479            lda  #0
     E423   A8           15480            tay
     E424   91 94        15481            sta  (dirbuf),y
     E426   C8           15482            iny
     E427   68           15483            pla
     E428   38           15484            sec
     E429   E9 01        15485            sbc  #1
     E42B   91 94        15486            sta  (dirbuf),y
     E42D   20 DE6C      15487            jsr  wrtss      ; write ss
     E430   20 D599      15488            jsr  watjob
     E433   20 EEF4      15489            jsr  mapout
     E436   20 CE0E      15490            jsr  fndrel
     E439   20 CF1E      15491            jsr  dblbuf     ; get back to leading buffer
     E43C   20 DEF8      15492            jsr  sspos
     E43F   70 03        15493            bvs  ar70
     E441   4C E275      15494            jmp  positn
     E444                15495   ar70
     E444   A9 80        15496            lda  #lrf
     E446   20 DD97      15497            jsr  setflg
     E449   A9 50        15498            lda  #norec
     E44B   20 C1C8      15499            jsr  cmderr
                         15500            .include  newss
```

```
Error Addr  Code           Seq    Source statement

                          15502            .subttl 'newss.src'
                          15503   ;*********************************
                          15504   ;* newss: generate new ss & fix  *
                          15505   ;*         old ss's to reflect it.*
                          15506   ;*   vars:                       *
                          15507   ;*   regs:                       *
                          15508   ;*                               *
                          15509   ;*********************************
                          15510
     E44E                 15511   newss
     E44E   20 F11E       15512           jsr  nxtts       ; get t&s based on hdr
     E451   20 CF1E       15513           jsr  dblbuf      ; use inactive buffer
     E454   20 DDF1       15514           jsr  scrub
     E457   20 DF93       15515           jsr  getact
     E45A   48            15516           pha
     E45B   20 DEC1       15517           jsr  clrbuf
     E45E   A6 82         15518           ldx  lindx
     E460   B5 CD         15519           lda  ss,x        ; set regs for transfer
     E462   A8            15520           tay
     E463   68            15521           pla
     E464   AA            15522           tax
     E465   A9 10         15523           lda  #ssioff     ; # of chars
     E467   20 DEA5       15524           jsr  b0tob0      ; transfer at buf(0)
                          15525
     E46A   A9 00         15526           lda  #0
     E46C   20 DEDC       15527           jsr  ssdir
     E46F   A0 02         15528           ldy  #2
     E471   B1 94         15529           lda  (dirbuf),y ; get ss #
     E473   48            15530           pha
     E474   A9 00         15531           lda  #0
     E476   20 D4C8       15532           jsr  setpnt
     E479   68            15533           pla
     E47A   18            15534           clc
     E47B   69 01         15535           adc  #1
     E47D   91 94         15536           sta  (dirbuf),y ; put ss # in new ss
     E47F   0A            15537           asl  a
     E480   69 04         15538           adc  #4
     E482   85 89         15539           sta  r3          ; save position
     E484   A8            15540           tay
     E485   38            15541           sec
     E486   E9 02         15542           sbc  #2
     E488   85 8A         15543           sta  r4
     E48A   A5 80         15544           lda  track
     E48C   85 87         15545           sta  r1          ; save for ss update
     E48E   91 94         15546           sta  (dirbuf),y ; put track in ss
     E490   C8            15547           iny
     E491   A5 81         15548           lda  sector
     E493   85 88         15549           sta  r2          ; save for ss update
     E495   91 94         15550           sta  (dirbuf),y ; put sector in ss
     E497   A0 00         15551           ldy  #0
```

```
Error Addr  Code           Seq    Source statement

     E499   98           15552            tya
     E49A   91 94        15553            sta  (dirbuf),y ; null link
     E49C   C8           15554            iny
     E49D   A9 11        15555            lda  #ssioff+1  ; ptr to last byte
     E49F   91 94        15556            sta  (dirbuf),y
                         15557
     E4A1   A9 10        15558            lda  #ssioff
     E4A3   20 D4C8      15559            jsr  setpnt
     E4A6   20 DE50      15560            jsr  wrtab
     E4A9   20 D599      15561            jsr  watjob
                         15562
     E4AC                15563   ns20
     E4AC   A6 82        15564            ldx  lindx
     E4AE   B5 CD        15565            lda  ss,x       ; get ss buffer #
     E4B0   48           15566            pha
     E4B1   20 DF9E      15567            jsr  gaflgs
     E4B4   A6 82        15568            ldx  lindx
     E4B6   95 CD        15569            sta  ss,x       ; swap act-buf & ss
     E4B8   68           15570            pla
     E4B9   AE 0257      15571            ldx  lbused
     E4BC   95 A7        15572            sta  buf0,x
                         15573
     E4BE   A9 00        15574            lda  #0
     E4C0   20 D4C8      15575            jsr  setpnt     ; set link to new ss
     E4C3   A0 00        15576            ldy  #0
     E4C5   A5 80        15577            lda  track
     E4C7   91 94        15578            sta  (dirbuf),y
     E4C9   C8           15579            iny
     E4CA   A5 81        15580            lda  sector
     E4CC   91 94        15581            sta  (dirbuf),y
     E4CE   4C E4DE      15582            jmp  ns50
     E4D1                15583   ns40
     E4D1   20 DF93      15584            jsr  getact
     E4D4   A6 82        15585            ldx  lindx
     E4D6   20 DF1B      15586            jsr  ibrd       ; read next ss
     E4D9   A9 00        15587            lda  #0
     E4DB   20 D4C8      15588            jsr  setpnt     ; ptr=0
     E4DE                15589   ns50
     E4DE   C6 8A        15590            dec  r4
     E4E0   C6 8A        15591            dec  r4
     E4E2   A4 89        15592            ldy  r3         ; get new ss link ptr
     E4E4   A5 87        15593            lda  r1
     E4E6   91 94        15594            sta  (dirbuf),y ; put track in
     E4E8   C8           15595            iny
     E4E9   A5 88        15596            lda  r2
     E4EB   91 94        15597            sta  (dirbuf),y ; put sector in
                         15598
     E4ED   20 DE5E      15599            jsr  wrtout     ; write it back...
     E4F0   20 D599      15600            jsr  watjob     ; ...& wait
     E4F3   A4 8A        15601            ldy  r4
```

```
Error Addr  Code          Seq    Source statement

     E4F5   C0 03        15602           cpy  #3
     E4F7   B0 D8        15603           bcs  ns40       ; more ss to update!
                         15604
     E4F9   4C CF1E      15605           jmp  dblbuf     ; reset active buffer
                         15606           .include  erproc
```

```
Error Addr  Code           Seq    Source statement

                          15608            .subttl 'erproc.src'
                          15609    ; error processing
                          15610
                          15611    ; controller errors
                          15612    ;  0  (1)  no error
                          15613    ; 20  (2)  can't find block header
                          15614    ; 21  (3)  no synch character
                          15615    ; 22  (4)  data block not present
                          15616    ; 23  (5)  checksum error in data
                          15617    ; 24  (16) byte decoding error
                          15618    ; 25  (7)  write-verify error
                          15619    ; 26  (8)  write w/ write protect on
                          15620    ; 27  (9)  checksum error in header
                          15621    ; 28  (10) data extends into next block
                          15622    ; 29  (11) disk i.d. mismatch
                          15623
                          15624    ; command errors
                          15625    ; 30  general syntax
                          15626    ; 31  invalid command
                          15627    ; 32  long line
                          15628    ; 33  invalid filname
                          15629    ; 34  no file given
                          15630    ; 39  command file not found
                          15631
                          15632    ; 50  record not present
                          15633    ; 51  overflow in record
                          15634    ; 52  file too large
                          15635
                          15636    ; 60  file open for write
                          15637    ; 61  file not open
                          15638    ; 62  file not found
                          15639    ; 63  file exists
                          15640    ; 64  file type mismatch
                          15641    ; 65  no block
                          15642    ; 66  illegal track or sector
                          15643    ; 67  illegal system t or s
                          15644
                          15645    ; 70  no channels available
                          15646    ; 71  directory error
                          15647    ; 72  disk full
                          15648    ; 73  cbm dos v3.0
                          15649    ; 74  drive not ready
                          15650
                          15651    ;  1  files scratched response
                          15652
              =0030       15653    badsyn   =$30
              =0031       15654    badcmd   =$31
              =0032       15655    longln   =$32
              =0033       15656    badfn    =$33
              =0034       15657    nofile   =$34
```

```
Error Addr  Code           Seq    Source statement

            =0039        15658   nocfil    =$39
            =0050        15659   norec     =$50
            =0051        15660   recovf    =$51
            =0052        15661   bigfil    =$52
            =0060        15662   filopn    =$60
            =0061        15663   filnop    =$61
            =0062        15664   flntfd    =$62
            =0063        15665   flexst    =$63
            =0064        15666   mistyp    =$64
            =0065        15667   noblk     =$65
            =0066        15668   badts     =$66
            =0067        15669   systs     =$67
            =0070        15670   nochnl    =$70
            =0071        15671   direrr    =$71
            =0072        15672   dskful    =$72
            =0073        15673   cbmv2     =$73
            =0074        15674   nodriv    =$74
```

```
Error Addr  Code           Seq   Source statement

                          15676   ; error message table
                          15677   ;   leading errror numbers,
                          15678   ;   text with 1st & last chars
                          15679   ;   or'ed with $80,
                          15680   ;   tokens for key words are
                          15681   ;   less than $10 (and'ed w/ $80)
                          15682
     E4FC                 15683   errtab          ; " OK"
     E4FC   00 A0 4F      15684           .byte    0,$a0,'O',$cb
     E4FF   CB
                          15685   ;"read error"
     E500   20 21 22      15686           .byte    $20,$21,$22,$23,$24,$27
     E503   23 24 27
     E506   D2 45 41      15687           .byte    $d2,'EAD',$89
     E509   44 89
                          15688   ;" file too large"
     E50B   52 83 20      15689           .byte    $52,$83,' TOO LARG',$c5
     E50E   54 4F 4F
     E511   20 4C 41
     E514   52 47 C5
                          15690   ;" record not present"
     E517   50 8B 06      15691           .byte    $50,$8b,6,' PRESEN',$d4
     E51A   20 50 52
     E51D   45 53 45
     E520   4E D4
                          15692   ;"overflow in record"
     E522   51 CF 56      15693           .byte    $51,$cf,'VERFLOW '
     E525   45 52 46
     E528   4C 4F 57
     E52B   20
     E52C   49 4E 8B      15694           .byte    'IN',$8b
                          15695   ;" write error"
     E52F   25 28 8A      15696           .byte    $25,$28,$8a,$89
     E532   89
                          15697   ;" write protect on"
     E533   26 8A 20      15698           .byte    $26,$8a,' PROTECT O',$ce
     E536   50 52 4F
     E539   54 45 43
     E53C   54 20 4F
     E53F   CE
                          15699   ;" disk id mismatch"
     E540   29 88 20      15700           .byte    $29,$88,' ID',$85
     E543   49 44 85
                          15701   ;"syntax error"
     E546   30 31 32      15702           .byte    $30,$31,$32,$33,$34
     E549   33 34
     E54B   D3 59 4E      15703           .byte    $d3,'YNTAX',$89
     E54E   54 41 58
     E551   89
                          15704   ;" write file open"
```

```
Error Addr  Code          Seq   Source statement

     E552   60 8A 03    15705           .byte    $60,$8a,3,$84
     E555   84
                        15706   ;" file exists"
     E556   63 83 20    15707           .byte    $63,$83,' EXIST',$d3
     E559   45 58 49
     E55C   53 54 D3
                        15708   ;" file type mismatch"
     E55F   64 83 20    15709           .byte    $64,$83,' TYPE',$85
     E562   54 59 50
     E565   45 85
                        15710   ;"no block"
     E567   65 CE 4F    15711           .byte    $65,$ce,'O BLOC',$cb
     E56A   20 42 4C
     E56D   4F 43 CB
                        15712   ;"illegal track or sector"
     E570   66 67 C9    15713           .byte   $66,$67,$c9,'LLEGAL TRACK'
     E573   4C 4C 45
     E576   47 41 4C
     E579   20 54 52
     E57C   41 43 4B
     E57F   20 4F 52    15714           .byte    ' OR SECTO',$d2
     E582   20 53 45
     E585   43 54 4F
     E588   D2
                        15715   ;" file not open"
     E589   61 83 06    15716           .byte    $61,$83,6,$84
     E58C   84
                        15717   ;" file not found"
     E58D   39 62 83    15718           .byte    $39,$62,$83,6,$87
     E590   06 87
                        15719   ;" files scratched"
     E592   01 83 53    15720           .byte    1,$83,'S SCRATCHE',$c4
     E595   20 53 43
     E598   52 41 54
     E59B   43 48 45
     E59E   C4
                        15721   ;"no channel"
     E59F   70 CE 4F    15722           .byte    $70,$ce,'O CHANNE',$cc
     E5A2   20 43 48
     E5A5   41 4E 4E
     E5A8   45 CC
                        15723   ;"dir error"
     E5AA   71 C4 49    15724           .byte    $71,$c4,'IR',$89
     E5AD   52 89
                        15725   ;" disk full"
     E5AF   72 88 20    15726           .byte    $72,$88,' FUL',$cc
     E5B2   46 55 4C
     E5B5   CC
                        15727   ;"cbm dos v3.0 1571"
     E5B6   73 C3 42    15728           .byte   $73,$c3,'BM DOS V3.0 157',$b1
```

```
Error Addr  Code           Seq    Source statement

     E5B9   4D 20 44
     E5BC   4F 53 20
     E5BF   56 33 2E
     E5C2   30 20 31
     E5C5   35 37 B1
                          15729   ;"drive not ready"
     E5C8   74 C4 52      15730           .byte   $74,$c4,'RIVE',6,' READ',$d9
     E5CB   49 56 45
     E5CE   06 20 52
     E5D1   45 41 44
     E5D4   D9
                          15731
                          15732   ; error token key words
                          15733   ;   words used more than once
                          15734   ;"error"
     E5D5   09 C5 52      15735           .byte    9,$c5,'RRO',$d2
     E5D8   52 4F D2
                          15736   ;"write"
     E5DB   0A D7 52      15737           .byte    $a,$d7,'RIT',$c5
     E5DE   49 54 C5
                          15738   ;"file"
     E5E1   03 C6 49      15739           .byte    3,$c6,'IL',$c5
     E5E4   4C C5
                          15740   ;"open"
     E5E6   04 CF 50      15741           .byte    4,$cf,'PE',$ce
     E5E9   45 CE
                          15742   ;"mismatch"
     E5EB   05 CD 49      15743           .byte    5,$cd,'ISMATC',$c8
     E5EE   53 4D 41
     E5F1   54 43 C8
                          15744   ;"not"
     E5F4   06 CE 4F      15745           .byte    6,$ce,'O',$d4
     E5F7   D4
                          15746   ;"found"
     E5F8   07 C6 4F      15747           .byte    7,$c6,'OUN',$c4
     E5FB   55 4E C4
                          15748   ;"disk"
     E5FE   08 C4 49      15749           .byte    8,$c4,'IS',$cb
     E601   53 CB
                          15750   ;"record"
     E603   0B D2 45      15751           .byte   $b,$d2,'ECOR',$c4
     E606   43 4F 52
     E609   C4
            =E60A         15752   etend    =*
```

```
Error Addr  Code          Seq   Source statement

                         15754   ; controller error entry
                         15755   ;   .a= error #
                         15756   ;   .x= job #
     E60A                15757   error
                         15758   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     E648                15804   cmder3
     E648   20 C1BD       15805           jsr  clrcb       ; clear cmdbuf
     E64B   A9 00         15806           lda  #0
     E64D   8D 02F9       15807           sta  wbam        ; clear after error
     E650   20 C12C       15808           jsr  erron       ; set error led
     E653   20 D4DA       15809           jsr  freich      ; free internal channel
     E656   A9 00         15810           lda  #0          ; clear pointers
     E658   85 A3         15811           sta  buftab+cbptr
     E65A   A2 45         15812           ldx  #topwrt
     E65C   9A            15813           txs              ;  purge stack
     E65D   A5 84         15814           lda  orgsa
     E65F   29 0F         15815           and  #$f
     E661   85 83         15816           sta  sa
     E663   C9 0F         15817           cmp  #$f
     E665   F0 31         15818           beq  err10
     E667   78            15819           sei
     E668   A5 79         15820           lda  lsnact
     E66A   D0 1C         15821           bne  lsnerr
     E66C   A5 7A         15822           lda  tlkact
     E66E   D0 10         15823           bne  tlkerr
                          15824
     E670   A6 83         15825           ldx  sa
     E672   BD 022B       15826           lda  lintab,x
     E675   C9 FF         15827           cmp  #$ff
     E677   F0 1F         15828           beq  err10
     E679   29 0F         15829           and  #$f
     E67B   85 82         15830           sta  lindx
     E67D   4C E68E       15831           jmp  tlerr
                          15832
                          15833
                          15834   ; talker error recovery
                          15835   ;  if command channel, release dav
                          15836   ;  if data channel, force not ready
                          15837   ;   and release channel
     E680                 15838   tlkerr
     E680   20 D0EB       15839           jsr  fndrch
                          15840   ;       jsr iterr               ; *** rom - 05 fix 8/18/83 ***
     E683   EA EA EA      15841           .byte  $ea,$ea,$ea      ; fill in 'jsr'
     E686   D0 06         15842           bne  tlerr              ; finish
                          15843
                          15844   ; listener error recovery
                          15845   ;  if command channel, release rfd
                          15846   ;  if data channel, force not ready
                          15847   ;  and release channel
     E688                 15848   lsnerr
     E688   20 D107       15849           jsr  fndwch
                          15850   ;       jsr ilerr               ; *** rom - 05 fix 8/18/83 ***
     E68B   EA EA EA      15851           .byte  $ea,$ea,$ea      ; fill in 'jsr'
     E68E                 15852   tlerr
     E68E   20 D125       15853           jsr  typfil
```

```
Error Addr  Code          Seq    Source statement

     E691   C9 04        15854            cmp  #reltyp
     E693   B0 03        15855            bcs  err10
     E695   20 D227      15856            jsr  frechn
     E698                15857    err10
                         15858
                         15859    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          15867   ; convert hex to bcd
      E69B  AA            15868   hexdec  tax
      E69C  A9 00         15869           lda  #0
      E69E  F8            15870           sed
      E69F  E0 00         15871   hex0    cpx  #0
      E6A1  F0 07         15872           beq  hex5
      E6A3  18            15873           clc
      E6A4  69 01         15874           adc  #1
      E6A6  CA            15875           dex
      E6A7  4C E69F       15876           jmp  hex0
      E6AA  D8            15877   hex5    cld
                          15878
                          15879   ; convert bcd to ascii dec
                          15880   ;  return bcd in .x
                          15881   ;  store ascii in (temp),y
      E6AB  AA            15882   bcddec  tax
      E6AC  4A            15883           lsr  a
      E6AD  4A            15884           lsr  a
      E6AE  4A            15885           lsr  a
      E6AF  4A            15886           lsr  a
      E6B0  20 E6B4       15887           jsr  bcd2
      E6B3  8A            15888           txa
      E6B4                15889   bcd2
      E6B4  29 0F         15890           and  #$f
      E6B6  09 30         15891           ora  #$30
      E6B8  91 A5         15892           sta  (cb+2),y
      E6BA  C8            15893           iny
      E6BB  60            15894           rts
                          15895
                          15896   ; transfer error message to
                          15897   ;  error buffer
                          15898
      E6BC                15899   okerr
      E6BC  20 C123       15900           jsr  erroff
      E6BF  A9 00         15901           lda  #0
      E6C1                15902   errts0
      E6C1  A0 00         15903           ldy  #0
      E6C3  84 80         15904           sty  track
      E6C5  84 81         15905           sty  sector
                          15906
      E6C7                15907   errmsg
      E6C7  A0 00         15908           ldy  #0
      E6C9  A2 D5         15909           ldx  #<errbuf
      E6CB  86 A5         15910           stx  cb+2
      E6CD  A2 02         15911           ldx  #>errbuf
      E6CF  86 A6         15912           stx  cb+3
      E6D1  20 E6AB       15913           jsr  bcddec     ; convert error #
      E6D4  A9 2C         15914           lda  #','
      E6D6  91 A5         15915           sta  (cb+2),y
      E6D8  C8            15916           iny
```

```
Error Addr  Code          Seq    Source statement

     E6D9   AD 02D5      15917           lda  errbuf
     E6DC   8D 0243      15918           sta  chndat+errchn
     E6DF   8A           15919           txa             ; error # in .x
     E6E0   20 E706      15920           jsr  ermove     ; move message
                         15921
     E6E3   A9 2C        15922   ermsg2  lda  #',
     E6E5   91 A5        15923           sta  (cb+2),y
     E6E7   C8           15924           iny
     E6E8   A5 80        15925           lda  track
     E6EA   20 E69B      15926           jsr  hexdec     ; convert track #
     E6ED   A9 2C        15927           lda  #',
     E6EF   91 A5        15928           sta  (cb+2),y
     E6F1   C8           15929           iny
     E6F2   A5 81        15930           lda  sector     ; convert sector #
     E6F4   20 E69B      15931           jsr  hexdec
     E6F7   88           15932           dey
     E6F8   98           15933           tya
     E6F9   18           15934           clc
     E6FA   69 D5        15935           adc  #<errbuf   ; set last char
     E6FC   8D 0249      15936           sta  lstchr+errchn
     E6FF   E6 A5        15937           inc  cb+2
     E701   A9 88        15938           lda  #rdytlk
     E703   85 F7        15939           sta  chnrdy+errchn
     E705   60           15940           rts
                         15941
                         15942   ;*********************************;
                         15943   ;*    ermove - move error message *;
                         15944   ;*      from errtab to errbuf.    *;
                         15945   ;*      fully recursive for token *;
                         15946   ;*      word prosessing.          *;
                         15947   ;*   input: .a= bcd error number  *;
                         15948   ;*********************************;
                         15949
     E706                15950   ermove
     E706   AA           15951           tax             ; save .a
     E707   A5 86        15952           lda  r0         ; save r0,r0+1
     E709   48           15953           pha
     E70A   A5 87        15954           lda  r0+1
     E70C   48           15955           pha
     E70D   A9 FC        15956           lda  #<errtab   ; set pointer to table
     E70F   85 86        15957           sta  r0
     E711   A9 E4        15958           lda  #>errtab
     E713   85 87        15959           sta  r0+1
     E715   8A           15960           txa             ; restore .a
     E716   A2 00        15961           ldx  #0         ; .x=0 for indirect
     E718                15962   e10
     E718   C1 86        15963           cmp  (r0,x)     ; ?error # = table entry?
     E71A   F0 21        15964           beq  e50        ; yes, send message
                         15965
     E71C   48           15966           pha             ; save error #
```

```
Error Addr  Code           Seq    Source statement

     E71D   20 E775      15967            jsr  eadv2       ; check & advance ptr
     E720   90 05        15968            bcc  e30         ; more #'s to check
     E722                15969    e20
     E722   20 E775      15970            jsr  eadv2       ; advance past this message
     E725   90 FB        15971            bcc  e20
     E727                15972    e30
     E727   A5 87        15973            lda  r0+1        ; check ptr
     E729   C9 E6        15974            cmp  #>etend
     E72B   90 08        15975            bcc  e40         ; <, continue
     E72D   D0 0A        15976            bne  e45         ; >, quit
                         15977
     E72F   A9 0A        15978            lda  #<etend
     E731   C5 86        15979            cmp  r0
     E733   90 04        15980            bcc  e45         ; past end of table
     E735                15981    e40
     E735   68           15982            pla              ; restore error #
     E736   4C E718      15983            jmp  e10         ; check next entry
     E739                15984    e45
     E739   68           15985            pla              ; pop error #
     E73A   4C E74D      15986            jmp  e90         ; go quit
                         15987
     E73D                15988    e50                      ; the number has been located
     E73D   20 E767      15989            jsr  eadv1
     E740   90 FB        15990            bcc  e50         ; advance past other #'s
     E742                15991    e55
     E742   20 E754      15992            jsr  e60
     E745   20 E767      15993            jsr  eadv1
     E748   90 F8        15994            bcc  e55
                         15995
     E74A   20 E754      15996            jsr  e60         ; check for token or last word
     E74D                15997    e90
     E74D   68           15998            pla              ; all finished
     E74E   85 87        15999            sta  r0+1        ; restore r0
     E750   68           16000            pla
     E751   85 86        16001            sta  r0
     E753   60           16002            rts
                         16003
     E754                16004    e60
     E754   C9 20        16005            cmp  #$20        ; (max token #)+1
     E756   B0 0B        16006            bcs  e70         ; not a token
     E758   AA           16007            tax
     E759   A9 20        16008            lda  #$20        ; implied leading space
     E75B   91 A5        16009            sta  (cb+2),y
     E75D   C8           16010            iny
     E75E   8A           16011            txa              ; restore token #
     E75F   20 E706      16012            jsr  ermove      ; add token word to message
     E762   60           16013            rts
     E763                16014    e70
     E763   91 A5        16015            sta  (cb+2),y    ; put char in message
     E765   C8           16016            iny
```

```
Error Addr  Code           Seq    Source statement

     E766   60           16017            rts
                         16018
                         16019    ;error advance & check
                         16020
     E767                16021    eadv1                       ; pre-increment
     E767   E6 86        16022             inc  r0            ; advance ptr
     E769   D0 02        16023             bne  ea10
     E76B   E6 87        16024             inc  r0+1
     E76D                16025    ea10
     E76D   A1 86        16026             lda  (r0,x)        ; get current entry
     E76F   0A           16027             asl  a             ; .c=1 is end or beginning
     E770   A1 86        16028             lda  (r0,x)
     E772   29 7F        16029             and  #$7f          ; mask off bit7
     E774   60           16030             rts
                         16031
     E775                16032    eadv2                       ; post-increment
     E775   20 E76D      16033             jsr  ea10          ; check table entry
     E778   E6 86        16034             inc  r0
     E77A   D0 02        16035             bne  ea20
     E77C   E6 87        16036             inc  r0+1
     E77E                16037    ea20
     E77E   60           16038             rts
                         16039             .include  utlodr
```

```
Error Addr  Code          Seq   Source statement

                        16041           .subttl 'utlodr.src'
                        16042   ;*********************************
                        16043   ;*utlodr-utility loader used to
                        16044   ; load user programs or system
                        16045   ; utilities from disk and
                        16046   ; execute them.
                        16047   ;
                        16048   ; this loader is designed to be
                        16049   ; incorporated into the 1540 dos
                        16050   ; system (serial bus interface).
                        16051   ;
                        16052   ; hardware reqired :
                        16053   ;  connect data and clock line to
                        16054   ;  ground. (2-4-5 on connector)
                        16055   ;
                        16056   ;*********************************
                        16057   ;*on entry-
                        16058   ; only requirement is that the
                        16059   ; filename of the file to be
                        16060   ; loaded be the first specified
                        16061   ; name in cmdbuf (the command
                        16062   ; buffer).
                        16063   ;
                        16064   ; registers: ignored
                        16065   ;
                        16066   ;*on exit-
                        16067   ; if the file existed on disk and
                        16068   ; could be found, and no checksum
                        16069   ; errors were encountered while
                        16070   ; loading it, it is now loaded
                        16071   ; into memory, ready to execute.
                        16072   ;
                        16073   ; registers: all destroyed
                        16074   ;
                        16075   ; execution of the loaded program
                        16076   ; is started at the first byte
                        16077   ; loaded.
                        16078   ;
                        16079   ; cmdbuf contains the parameter
                        16080   ; string for the freshly loaded
                        16081   ; utility or user program.
                        16082   ;
                        16083   ;*********************************
```

```
Error Addr  Code           Seq    Source statement

                         16085    ;*********************************
                         16086    ;
                         16087    ;*first writing-
                         16088    ; 25-jan-80
                         16089    ; by ron stephens
                         16090    ;
                         16091    ;---------------------------------
                         16092    ;
                         16093    ;*revision list-
                         16094    ; 28-feb-80  s. patterson - add parser interface
                         16095    ; 09-apr-81 rsr - change for serial bus
                         16096    ; 01-jan-85 ds - fix utlodr filename parser
                         16097    ;
                         16098    ;*********************************
```

```
Error Addr  Code          Seq   Source statement

                        16100   ;*********************************
                        16101   ;
                        16102   ;*routines external to this
                        16103   ; module that are used:
                        16104   ;
                        16105   ; lookup-
                        16106   ;   function-finds first name
                        16107   ;   specified in cmdbuf in the
                        16108   ;   disk directory on the specified
                        16109   ;   drive.
                        16110   ;
                        16111   ;   inputs from utlodr-none
                        16112   ;   outputs to utlodr-none
                        16113   ;
                        16114   ; chkin-
                        16115   ;   function-checks if name was
                        16116   ;   found by lookup. error if not
                        16117   ;   found. exits to 8050 error
                        16118   ;   routine.
                        16119   ;
                        16120   ;   inputs from utlodr-none
                        16121   ;   outputs to utlodr-none
                        16122   ;
                        16123   ; gibyte-
                        16124   ;   function-fetches next byte
                        16125   ;   from open file. also sets
                        16126   ;   eoiflg zero if end of file
                        16127   ;   condition detected.
                        16128   ;
                        16129   ;   inputs from utlodr-none
                        16130   ;   outputs to utlodr-variable 'data' contains
                        16131   ;   the data byte.
                        16132   ;
                        16133   ; opntyp-
                        16134   ;   function-opens file previously
                        16135   ;   "looked up" by lookup above.
                        16136   ;   calls error routine if
                        16137   ;   type doesn't match that specified
                        16138   ;
                        16139   ;   inputs from utlodr-.a=file type
                        16140   ;   (5 in our case)
                        16141   ;   outputs to utlodr-none
                        16142   ;
                        16143   ; cmder2-
                        16144   ;   function-places error message
                        16145   ;   specified into error buffer.
                        16146   ;
                        16147   ;   inputs from utlodr-.a=error number
                        16148   ;   outputs to utlodr-none
                        16149   ;
```

```
Error Addr  Code          Seq    Source statement

                        16150   ;*********************************
                        16151   ; boot routine removed, won't work with serial i/f
```

```
Error Addr  Code           Seq   Source statement

     E77F               16153   boot2
     E77F   60          16154           rts     ; exit
                        16155   ;
                        16156   ;***rom -05 fix 8/18/83
     E780               16157   boot            ; power-on diag sense loader
     E780               16170   boot3
                        16182   ;---------rom -05 fix 8/18/83---------------
     E780   60 EA EA    16183           .byte   $60,$ea,$ea,$ea,$ea,$ea,$ea,$ea
     E783   EA EA EA
     E786   EA EA
     E788   EA EA EA    16184           .byte   $ea,$ea,$ea,$ea,$ea,$ea,$ea,$ea
     E78B   EA EA EA
     E78E   EA EA
     E790   EA EA EA    16185           .byte   $ea,$ea,$ea,$ea,$ea,$ea,$ea,$ea
     E793   EA EA EA
     E796   EA EA
     E798   EA EA EA    16186           .byte   $ea,$ea,$ea,$ea,$ea,$ea,$ea,$ea
     E79B   EA EA EA
     E79E   EA EA
     E7A0   EA EA 60    16187           .byte   $ea,$ea,$60
                        16188   ;-----------------------------------------
                        16189   ;
                        16190   ;
                        16191   ;*entry point
                        16192   ;
     E7A3               16193   utlodr
                        16194
                        16195   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                        16196
     E7A3   4C A5FE     16197           jmp  ptch18     ; set for filename *rom-05ds 01/22/85*
     E7A6   EA          16198           nop             ; fill
     E7A7   EA          16199           nop             ; fill
                        16200
                        16201   ;       lda  #$8d
                        16202   ;       jsr  parse
     E7A8               16203   rtch18
                        16204
                        16205   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                        16206
     E7A8               16207   boot4
     E7A8   20 F258     16208           jsr  killp      ; kill protect
     E7AB   AD 0278     16209           lda  f2cnt
     E7AE   48          16210           pha             ; save file count for utility
     E7AF   A9 01       16211           lda  #1
     E7B1   8D 0278     16212           sta  f2cnt
     E7B4   A9 FF       16213           lda  #$ff       ; init firstbyte flag
     E7B6   85 86       16214           sta  r0         ; r0 is flag
                        16215
     E7B8   20 C44F     16216           jsr  lookup     ; locate filename on disk
                        16217
```

```
Error Addr  Code           Seq    Source statement

     E7BB   AD 0280      16218            lda  filtrk     ; check if found. err if not
     E7BE   D0 05        16219            bne  utld00
     E7C0   A9 39        16220            lda  #nocfil
     E7C2   20 C1C8      16221            jsr  cmderr
     E7C5                16222   utld00
     E7C5   68           16223            pla
     E7C6   8D 0278      16224            sta  f2cnt      ; restore file count
                         16225
     E7C9   AD 0280      16226            lda  filtrk     ; init trk, sctr for open
     E7CC   85 80        16227            sta  track
     E7CE   AD 0285      16228            lda  filsec
     E7D1   85 81        16229            sta  sector
                         16230
     E7D3   A9 03        16231            lda  #usrtyp    ; open system type file( 5 )
     E7D5   20 D477      16232            jsr  opntyp     ; open
                         16233   ;
                         16234   ;*********************************
                         16235   ;
                         16236   ;*file record fetch loop
     E7D8   A9 00        16237   utld10   lda  #$00       ; init checksum
     E7DA   85 87        16238            sta  r1         ; calc. chksum resides in r1
                         16239
     E7DC   20 E839      16240            jsr  gtabyt     ; fetch load address lo
     E7DF   85 88        16241            sta  r2
     E7E1   20 E84B      16242            jsr  addsum     ; add into checksum
                         16243
     E7E4   20 E839      16244            jsr  gtabyt     ; fetch load address hi
     E7E7   85 89        16245            sta  r3
     E7E9   20 E84B      16246            jsr  addsum
                         16247
     E7EC   A5 86        16248            lda  r0         ; is this the firstbyte address?
     E7EE   F0 0A        16249            beq  utld20     ; br if not
     E7F0   A5 88        16250            lda  r2         ; sav away this addr. in2 stack
     E7F2   48           16251            pha             ; lo first
     E7F3   A5 89        16252            lda  r3
     E7F5   48           16253            pha             ; hi next
     E7F6   A9 00        16254            lda  #$00       ; clear flag
     E7F8   85 86        16255            sta  r0         ; firstbyte flag
                         16256
     E7FA   20 E839      16257   utld20   jsr  gtabyt     ; fetch data byte count
     E7FD   85 8A        16258            sta  r4         ; save in r4
     E7FF   20 E84B      16259            jsr  addsum     ; add into checksum
                         16260   ;
                         16261   ;*********************************
                         16262   ;
                         16263   ;*byte store loop
     E802   20 E839      16264   utld30   jsr  gtabyt     ; fetch data byt
     E805   A0 00        16265            ldy  #$00       ; init index
     E807   91 88        16266            sta  (r2),y     ; store byte at desired address
     E809   20 E84B      16267            jsr  addsum     ; add into checksum
```

```
Error Addr  Code           Seq    Source statement

                          16268
     E80C   A5 88         16269            lda  r2          ; pointer:=pointer+1
     E80E   18            16270            clc
     E80F   69 01         16271            adc  #$01
     E811   85 88         16272            sta  r2
     E813   90 02         16273            bcc  utld35
     E815   E6 89         16274            inc  r3          ; add in carry
                          16275
     E817   C6 8A         16276   utld35   dec  r4          ; update byte counter
     E819   D0 E7         16277            bne  utld30      ; if nonzero, continue
                          16278
                          16279   ;*end of byte store loop
                          16280   ;
                          16281   ;*********************************
                          16282   ;
     E81B   20 CA35       16283            jsr  gibyte      ; get a byte without check for eoi
     E81E   A5 85         16284            lda  data
     E820   C5 87         16285            cmp  r1          ; last byte fetched was chksum
     E822   F0 08         16286            beq  utld50      ; if same...everything ok
     E824   20 DE3E       16287            jsr  gethdr
     E827   A9 50         16288            lda  #norec      ; show record overflow
     E829   20 E645       16289            jsr  cmder2      ; and leave to error exit
                          16290
     E82C   A5 F8         16291   utld50   lda  eoiflg     ; check for end of file
     E82E   D0 A8         16292            bne  utld10      ; if nonzero, not done
                          16293   ;*end of record load loop
                          16294   ;
                          16295   ;*********************************
                          16296   ;
     E830   68            16297            pla              ; xfer cntrl to 1st byte addr.
     E831   85 89         16298            sta  r3
     E833   68            16299            pla
     E834   85 88         16300            sta  r2
     E836   6C 0088       16301            jmp  (r2)
```

```
Error Addr  Code           Seq    Source statement

                          16303   ;*********************************
                          16304   ;
                          16305   ;*local routines used by utlodr
                          16306   ;
                          16307   ;*********************************
                          16308   ;
                          16309   ;*gtabyt-fetches a byte from the
                          16310   ;        file open on the internal
                          16311   ;        channel. checks if this
                          16312   ;        was the last byte in the
                          16313   ;        file. error if it was.
                          16314   ;        show a 'pter' (premature
                          16315   ;        termination error).
                          16316   ;
                          16317   ;*********************************
                          16318   ;
                          16319   ;*entry point
                          16320   ;
     E839   20 CA35       16321   gtabyt  jsr  gibyte      ; fetch a byte to data loc
     E83C   A5 F8         16322           lda  eoiflg      ; check if eof exists
     E83E   D0 08         16323           bne  gtabye      ; ok if nonzero
     E840   20 DE3E       16324           jsr  gethdr
     E843   A9 51         16325           lda  #recovf     ; record size error
     E845   20 E645       16326           jsr  cmder2      ; call 8050 error routine
                          16327
     E848   A5 85         16328   gtabye  lda  data
     E84A   60            16329           rts
                          16330
                          16331   ;*********************************
                          16332   ;
                          16333   ;*addsum-adds up checksum into
                          16334   ;        location r1. algorithm:
                          16335   ;        newsum:=oldsum+newbyte+
                          16336   ;        carry
                          16337   ;
                          16338   ; inputs: expects newbyte in .a
                          16339   ; outputs: r1=newsum, .a destroyed
                          16340   ;
                          16341   ;*********************************
                          16342   ;
                          16343   ;*entry point
                          16344   ;
     E84B   18            16345   addsum  clc
     E84C   65 87         16346           adc  r1          ; .a=.a+r1
     E84E   69 00         16347           adc  #$00        ; .a=.a+carry
     E850   85 87         16348           sta  r1          ; save new checksum
     E852   60            16349           rts
                          16350           .include  seratn
```

```
Error Addr  Code           Seq    Source statement

                          16352            .subttl 'ser.atn'
                          16353
     E853   AD 1801       16354   atnirq   lda  pa1         ;  clear irq (ca1)
     E856   A9 01         16355            lda  #1
     E858   85 7C         16356            sta  atnpnd
     E85A   60            16357            rts
                          16358
     E85B   78            16359   atnsrv   sei
                          16360
     E85C   A9 00         16361            lda  #0          ;  clear pending flag
     E85E   85 7C         16362            sta  atnpnd
     E860   85 79         16363            sta  lsnact      ;  clear address flags
     E862   85 7A         16364            sta  tlkact
                          16365
     E864   A2 45         16366            ldx  #topwrt     ;  reset stack
     E866   9A            16367            txs
                          16368
     E867   A9 80         16369            lda  #$80        ;  set atn mode flag for acpt routine
     E869   85 F8         16370            sta  eoiflg      ; reset eoi flag to non-eoi state
     E86B   85 7D         16371            sta  atnmod
                          16372
     E86D   20 E9B7       16373            jsr  clkhi
     E870   20 E9A5       16374            jsr  datlow      ;  set data line low as response
                          16375
     E873   AD 1800       16376            lda  pb
                          16377   ; set atn ack to get hardware control of data line
     E876   09 10         16378            ora  #atna
     E878   8D 1800       16379            sta  pb
                          16380
     E87B   AD 1800       16381   atns15   lda  pb          ;  test atn still here
     E87E   10 57         16382            bpl  atns20      ;  gone !
     E880   29 04         16383            and  #clkin      ;  clock still low
     E882   D0 F7         16384            bne  atns15
                          16385
                          16386
     E884   20 E9C9       16387   atn30    jsr  acptr       ;  get a command byte
                          16388
     E887   C9 3F         16389            cmp  #unlsn
     E889   D0 06         16390            bne  atn35
                          16391
     E88B   A9 00         16392            lda  #0          ;  general unlisten command
     E88D   85 79         16393            sta  lsnact
     E88F   F0 71         16394            beq  atn122      ;  jmp
                          16395
     E891   C9 5F         16396   atn35    cmp  #untlk
     E893   D0 06         16397            bne  atn40
                          16398
     E895   A9 00         16399            lda  #0
     E897   85 7A         16400            sta  tlkact      ;  untalk now
     E899   F0 67         16401            beq  atn122      ;  jmp
```

```
Error Addr  Code           Seq    Source statement

                          16402
     E89B   C5 78         16403   atn40   cmp  tlkadr     ;  our talk address?
     E89D   DO OA         16404           bne  atn45      ;  nope
                          16405
     E89F   A9 01         16406           lda  #1         ;  yes
     E8A1   85 7A         16407           sta  tlkact     ;  set active talker flag
     E8A3   A9 00         16408           lda  #0
     E8A5   85 79         16409           sta  lsnact     ;  unlisten
     E8A7   F0 29         16410           beq  atn95      ; jump
                          16411
     E8A9   C5 77         16412   atn45   cmp  lsnadr     ;  out listen address?
     E8AB   DO OA         16413           bne  atn50      ;  nope
                          16414
     E8AD   A9 01         16415           lda  #1         ;  yes
     E8AF   85 79         16416           sta  lsnact
     E8B1   A9 00         16417           lda  #0
     E8B3   85 7A         16418           sta  tlkact
     E8B5   F0 1B         16419           beq  atn95      ; jump
                          16420
     E8B7   AA            16421   atn50   tax            ;  test if sa
     E8B8   29 60         16422           and  #$60
     E8BA   C9 60         16423           cmp  #$60       ;  sa = $60 + n
     E8BC   DO 3F         16424           bne  atn120     ; did not get a valid command
                          16425
                          16426
     E8BE   8A            16427           txa            ;  a sa for me
     E8BF   85 84         16428           sta  orgsa
     E8C1   29 OF         16429           and  #$0f       ;  strip junk
     E8C3   85 83         16430           sta  sa
                          16431
     E8C5   A5 84         16432           lda  orgsa      ;  test if close
     E8C7   29 F0         16433           and  #$f0
     E8C9   C9 E0         16434           cmp  #$e0
     E8CB   DO 35         16435           bne  atn122     ;  no
                          16436
     E8CD   58            16437           cli
     E8CE   20 DACO       16438           jsr  close      ;  close the file
     E8D1   78            16439           sei
                          16440
                          16441   ; warning:::close doesn't return in time for atn122:::
                          16442
     E8D2   2C 1800       16443   atn95   bit  pb         ;  test atn still here
     E8D5   30 AD         16444           bmi  atn30
                          16445
                          16446
                          16447   ; atn gone , do what we where told to do
                          16448
                          16449
     E8D7   A9 00         16450   atns20  lda  #0         ;  clear atn mode
     E8D9   85 7D         16451           sta  atnmod
```

```
Error Addr  Code           Seq    Source statement

                          16452
     E8DB   AD 1800       16453            lda  pb          ;  atn gone , release atn ack
     E8DE   29 EF         16454            and  #$ff-atna
     E8E0   8D 1800       16455            sta  pb
                          16456
     E8E3   A5 79         16457            lda  lsnact      ;  listen ?
     E8E5   F0 06         16458            beq  atn100
                          16459
     E8E7   20 EA2E       16460            jsr  listen
     E8EA   4C 836B       16461            jmp  xidle
                          16462
     E8ED   A5 7A         16463   atn100   lda  tlkact      ;  talk?
     E8EF   F0 09         16464            beq  atn110
                          16465
     E8F1   20 E99C       16466            jsr  dathi       ;  release data line
     E8F4   20 E9AE       16467            jsr  clklow      ;  pull clock low
                          16468
     E8F7   20 E909       16469            jsr  talk
     E8FA   4C EA4E       16470   atn110   jmp  ilerr       ;  release all lines and go to xidle
                          16471
                          16472   ; fix so (device not present) errors reported
                          16473
     E8FD   A9 10         16474   atn120   lda  #$10        ; kill all but atnak
     E8FF   8D 1800       16475            sta  pb
     E902   2C 1800       16476   atn122   bit  pb
     E905   10 D0         16477            bpl  atns20      ; exit out same way after atn done
     E907   30 F9         16478            bmi  atn122      ; jmp
                          16479
                          16480   ; rsr 12/8/80 fix inverted logic
                          16481   ; rsr 12/16/80 fix bad logic
                          16482   ; rsr 12/31/80 fix eoiflg and logic
                          16483            .include  sertalk
```

```
Error Addr  Code           Seq    Source statement

                          16485            .subttl 'ser.talk'
     E909   78            16486    talk    sei             ;  find if open channel
     E90A   20 DOEB       16487            jsr  fndrch
     E90D   BO 06         16488            bcs  notlk      ;  no one home
                          16489
     E90F   A6 82         16490    talk1   ldx  lindx
     E911   B5 F2         16491            lda  chnrdy,x
     E913   30 01         16492            bmi  tlk05
                          16493
     E915   60            16494    notlk   rts
                          16495
                          16496    ;  code added to correct verify error
     E916   20 EA59       16497    tlk05   jsr  tstatn     ; test for atn
     E919   20 E9CO       16498            jsr  debnc
                          16499
                          16500    ;check for clock gone (must be 80us+ from jmp talk1)
                          16501
     E91C   29 01         16502            and  #datin
     E91E   08            16503            php
     E91F   20 E9B7       16504            jsr  clkhi      ; set clock line hi
     E922   28            16505            plp             ; see if verify error...
     E923   FO 12         16506            beq  tlk02      ; yes...data line low, eoi !!!!
                          16507
     E925   20 EA59       16508    talk2   jsr  tstatn     ;  test atn gone
     E928   20 E9CO       16509            jsr  debnc      ;  test data line now low
     E92B   29 01         16510            and  #datin
     E92D   DO F6         16511            bne  talk2      ;  did not respond
                          16512
                          16513
     E92F   A6 82         16514            ldx  lindx      ;  prepare to send eoi if needed
     E931   B5 F2         16515            lda  chnrdy,x
     E933   29 08         16516            and  #eoi
     E935   DO 14         16517            bne  noeoi      ;  no eoi
                          16518
     E937   20 EA59       16519    tlk02   jsr  tstatn
     E93A   20 E9CO       16520            jsr  debnc      ;  send an eoi
     E93D   29 01         16521            and  #datin     ;  test if data line is low
     E93F   DO F6         16522            bne  tlk02      ;  yes, wait till hi
                          16523
     E941   20 EA59       16524    tlk03   jsr  tstatn
     E944   20 E9CO       16525            jsr  debnc      ;  wait for data line to go low
     E947   29 01         16526            and  #datin
     E949   FO F6         16527            beq  tlk03
                          16528
     E94B   20 E9AE       16529    noeoi   jsr  clklow     ;  set clock low
     E94E   20 EA59       16530            jsr  tstatn
     E951   20 E9CO       16531            jsr  debnc      ;  wait for data  line hi
     E954   29 01         16532            and  #datin
     E956   DO F3         16533            bne  noeoi
                          16534
```

```
Error Addr  Code           Seq    Source statement

     E958   A9 08        16535            lda  #8          ;  set up bit counter
     E95A   85 98        16536            sta  cont
                         16537
     E95C   20 E9C0      16538    isr01   jsr  debnc       ;  let port settle
                         16539
     E95F   29 01        16540            and  #datin
                         16541    ; test that data line is now high before we send
     E961   D0 36        16542            bne  frmerx
                         16543
     E963   A6 82        16544    isr02   ldx  lindx       ;  get byte to send
     E965   BD 023E      16545            lda  chndat,x
     E968   6A           16546            ror  a
     E969   9D 023E      16547            sta  chndat,x
                         16548
     E96C   B0 05        16549            bcs  isrhi       ;  send a 1
                         16550
     E96E   20 E9A5      16551            jsr  datlow      ;  send a 0
     E971   D0 03        16552            bne  isrclk      ;  and clock it
                         16553
     E973   20 E99C      16554    isrhi   jsr  dathi
                         16555
     E976   20 E9B7      16556    isrclk  jsr  clkhi       ;  rising edge clock
                         16557
     E979   A5 23        16558            lda  drvtrk+1    ; check speed flag
     E97B   D0 03        16559            bne  isr03       ; no slow down
                         16560
     E97D   20 FEF3      16561            jsr  slowd       ;  slow down
     E980   20 FEFB      16562    isr03   jsr  clkdat      ;  pull clock low and release data
                         16563
     E983   C6 98        16564            dec  cont        ;  more bits?
     E985   D0 D5        16565            bne  isr01       ;  yes
                         16566
     E987   20 EA59      16567    isr04   jsr  tstatn
     E98A   20 E9C0      16568            jsr  debnc
     E98D   29 01        16569            and  #datin
     E98F   F0 F6        16570            beq  isr04
                         16571
     E991   58           16572            cli              ;  get next byte to send
     E992   20 D3AA      16573            jsr  get
     E995   78           16574            sei
                         16575
     E996   4C E90F      16576            jmp  talk1       ;  keep on talkin
                         16577
     E999   4C EA4E      16578    frmerx  jmp  frmerr
                         16579
                         16580
     E99C   AD 1800      16581    dathi   lda  pb          ;  set data out hi
     E99F   29 FD        16582            and  #$ff-datout
     E9A1   8D 1800      16583            sta  pb
     E9A4   60           16584            rts
```

```
Error Addr  Code           Seq    Source statement

                          16585
                          16586
                          16587
     E9A5   AD 1800       16588   datlow  lda  pb
     E9A8   09 02         16589           ora  #datout
     E9AA   8D 1800       16590           sta  pb
     E9AD   60            16591           rts
                          16592
                          16593
                          16594
     E9AE   AD 1800       16595   clklow  lda  pb
     E9B1   09 08         16596           ora  #clkout
     E9B3   8D 1800       16597           sta  pb
     E9B6   60            16598           rts
                          16599
                          16600
                          16601
     E9B7   AD 1800       16602   clkhi   lda  pb
     E9BA   29 F7         16603           and  #$ff-clkout
     E9BC   8D 1800       16604           sta  pb
     E9BF   60            16605           rts
                          16606
                          16607
                          16608
     E9C0   AD 1800       16609   debnc   lda  pb
     E9C3   CD 1800       16610           cmp  pb
     E9C6   D0 F8         16611           bne  debnc
     E9C8   60            16612           rts
                          16613
                          16614   ; additions after code sent to japan.
                          16615   ; rsr 3/30/81 add verify error correction
                          16616   ; rsr 3/31/81 change ordering to detect verify error
                          16617           .include  serlistn
```

```
Error Addr  Code           Seq    Source statement

                          16619            .subttl 'serlistn.src'
      E9C9   A9 08        16620    acptr   lda  #8          ;  set byte bit count
      E9CB   85 98        16621            sta  cont
                          16622
      E9CD   20 EA59      16623    acp00a  jsr  tstatn
      E9D0   20 E9C0      16624            jsr  debnc
      E9D3   29 04        16625            and  #clkin
      E9D5   D0 F6        16626            bne  acp00a
                          16627
      E9D7   20 E99C      16628            jsr  dathi       ;  make data line hi
                          16629
      E9DA   A9 01        16630            lda  #1          ;  wait 255 us
                          16631
                          16632    ;------rom -05 8/18/83-------------
      E9DC   4C FF20      16633            jmp  patch6
                          16634    ;---------------------------------
                          16635
      E9DF   20 EA59      16636    acp00   jsr  tstatn
      E9E2   AD 180D      16637            lda  ifr1
      E9E5   29 40        16638            and  #$40        ;  test if time out
      E9E7   D0 09        16639            bne  acp00b      ;  ran out,its an eoi
                          16640
      E9E9   20 E9C0      16641            jsr  debnc       ;  test clock low
      E9EC   29 04        16642            and  #clkin
      E9EE   F0 EF        16643            beq  acp00       ;  no
                          16644
      E9F0   D0 19        16645            bne  acp01       ;  yes
                          16646
      E9F2   20 E9A5      16647    acp00b  jsr  datlow      ;  set data line low as response
                          16648
      E9F5   A2 0A        16649            ldx  #10         ;  delay for talker turnaround
      E9F7   CA           16650    acp02   dex
      E9F8   D0 FD        16651            bne  acp02
                          16652
      E9FA   20 E99C      16653            jsr  dathi       ;  set data line hi
                          16654
      E9FD   20 EA59      16655    acp02a  jsr  tstatn
      EA00   20 E9C0      16656            jsr  debnc       ;  wait for low clock
      EA03   29 04        16657            and  #clkin
      EA05   F0 F6        16658            beq  acp02a
                          16659
      EA07   A9 00        16660            lda  #0          ;  set eoi received
      EA09   85 F8        16661            sta  eoiflg
                          16662
      EA0B                16663    acp01
      EA0B   AD 1800      16664    acp03   lda  pb          ;  wait for clock high
      EA0E   49 01        16665            eor  #01         ; complement datain
      EA10   4A           16666            lsr  a           ; shift into carry
      EA11   29 02        16667            and  #$02        ; clkin/2
      EA13   D0 F6        16668            bne  acp03
```

```
Error Addr  Code           Seq    Source statement

                          16669
      EA15  EA            16670            nop             ; fill space left by speed-up
      EA16  EA            16671            nop             ; to fix pal vc20
      EA17  EA            16672            nop             ;  901229-02 rom
      EA18  66 85         16673            ror  data
                          16674
      EA1A  20 EA59       16675   acp03a   jsr  tstatn
      EA1D  20 E9C0       16676            jsr  debnc
      EA20  29 04         16677            and  #clkin     ;  wait for clock low
      EA22  F0 F6         16678            beq  acp03a
                          16679
      EA24  C6 98         16680            dec  cont       ;  more to do?
      EA26  D0 E3         16681            bne  acp03
                          16682
      EA28  20 E9A5       16683            jsr  datlow     ;  set data line low
      EA2B  A5 85         16684            lda  data
      EA2D  60            16685            rts
                          16686
                          16687
                          16688
      EA2E  78            16689   listen   sei
                          16690
      EA2F  20 D107       16691            jsr  fndwch     ;  test if active write channel
      EA32  B0 05         16692            bcs  lsn15
                          16693
      EA34  B5 F2         16694            lda  chnrdy,x
      EA36  6A            16695            ror  a
      EA37  B0 0B         16696            bcs  lsn30
                          16697
      EA39  A5 84         16698   lsn15    lda  orgsa      ;  test if open
      EA3B  29 F0         16699            and  #$f0
      EA3D  C9 F0         16700            cmp  #$f0
      EA3F  F0 03         16701            beq  lsn30      ;  its an open
                          16702
      EA41  4C EA4E       16703            jmp  ilerr      ;  not active channel
                          16704
      EA44  20 E9C9       16705   lsn30    jsr  acptr      ;  get a byte
                          16706
      EA47  58            16707            cli
      EA48  20 CFB7       16708            jsr  put        ;  put(data,eoiflg,sa)
                          16709
      EA4B  4C EA2E       16710            jmp  listen     ;  and keep on listen
                          16711
                          16712
      EA4E                16713   frmerr
      EA4E                16714   iterr
      EA4E  A9 00         16715   ilerr    lda  #00        ;  release all bus lines
      EA50  8D 1800       16716            sta  pb
      EA53  4C 836B       16717            jmp  xidle
                          16718
```

```
Error Addr  Code          Seq    Source statement

                         16719
                         16720
     EA56   4C E85B      16721   atnlow  jmp  atnsrv
                         16722
                         16723
                         16724   ; tstatn()
                         16725   ;
                         16726   ;  if(atnmod)
                         16727   ;
                         16728   ;   if(pb & $80)
                         16729   ;      if 1571 then jatns20() else atns20
                         16730   ;
                         16731   ;   else return
                         16732   ;
                         16733   ;  else
                         16734   ;
                         16735   ;   if(pb&$80) return
                         16736   ;   else
                         16737   ;       if 1571 then jatnsrv() else atnsrv()
                         16738   ;
                         16739   ;
                         16740   ;
                         16741
                         16742
     EA59   A5 7D        16743   tstatn  lda  atnmod     ;  test if in atn mode
     EA5B   F0 06        16744           beq  tsta50     ;  no
                         16745
     EA5D   AD 1800      16746           lda  pb         ;  in atnmod
     EA60   10 09        16747           bpl  tatn20     ;  atn gone,do what we are told to do
                         16748
     EA62   60           16749   tstrtn  rts             ;  still in atn  mode
                         16750
     EA63   AD 1800      16751   tsta50  lda  pb         ;  not atnmode
     EA66   10 FA        16752           bpl  tstrtn     ;  no atn present
                         16753
                         16754   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          16765            .subttl 'dskintsf.src'
                          16766
                          16767    ; error display routine
                          16768    ; blinks the (error #)+1 in all three leds
                          16769
     EA6E   A2 00         16770    pezro   ldx  #0          ; error #1 for zero page
     EA70   2C            16771            .byte skip2     ; skip next two bytes
     EA71   A6 6F         16772    perr    ldx  temp        ; get error #
     EA73   9A            16773            txs              ; use stack as storage reg.
     EA74   BA            16774    pe20    tsx              ; restore error #
     EA75   A9 08         16775    pe30    lda  #led0+led1
     EA77   0D 1C00       16776            ora  ledprt
     EA7A   4C FEEA       16777            jmp  pea7a
                          16778
                          16779    ; turn on led !!!!patch so ddrb led is output!!!!
                          16780
     EA7D   98            16781    rea7d   tya              ; clear inner ctr !!!!patch return!!!!
     EA7E   18            16782.   pd10    clc
     EA7F   69 01         16783    pd20    adc  #1          ; count inner ctr
     EA81   D0 FC         16784            bne  pd20
     EA83   88            16785            dey              ; done ?
     EA84   D0 F8         16786            bne  pd10        ; no
                          16787
     EA86   AD 1C00       16788            lda  ledprt
     EA89   29 F7         16789            and  #$ff-led0-led1
     EA8B   8D 1C00       16790            sta  ledprt      ; turn off all leds
     EA8E                 16791    pe40                     ; wait
     EA8E   98            16792            tya              ; clear inner ctr
     EA8F   18            16793    pd11    clc
     EA90   69 01         16794    pd21    adc  #1          ; count inner ctr
     EA92   D0 FC         16795            bne  pd21
     EA94   88            16796            dey              ; done ?
     EA95   D0 F8         16797            bne  pd11        ; no
                          16798
     EA97   CA            16799            dex              ; blinked # ?
     EA98   10 DB         16800            bpl  pe30        ; no - blink again
     EA9A   E0 FC         16801            cpx  #$fc        ; waited between counts ?
     EA9C   D0 F0         16802            bne  pe40        ; no
     EA9E   F0 D4         16803            beq  pe20        ; always - all again
                          16804
     EAA0   78            16805    dskint  sei
     EAA1   D8            16806            cld
     EAA2   A2 66         16807            ldx  #$66        ; *,atnout,clk,*,*,side,fsdir,trk0
     EAA4   4C FF10       16808            jmp  patch5      ; *** rom ds 8/18/83 ***
     EAA7   E8            16809    dkit10  inx              ; fill
                          16810
                          16811    ;*********************************
                          16812    ;
                          16813    ; power up diagnostic
                          16814    ;
```

```
Error Addr  Code           Seq    Source statement

                          16815  ;*********************************
                          16816
     EAA8   AO OO         16817          ldy  #0
     EAAA   A2 OO         16818          ldx  #0
     EAAC   8A            16819  pu10    txa              ; fill z-page accend pattern
     EAAD   95 OO         16820          sta  $0,x
     EAAF   E8            16821          inx
     EABO   DO FA         16822          bne  pu10
     EAB2   8A            16823  pu20    txa              ; check pattern by inc...
     EAB3   D5 OO         16824          cmp  $0,x        ; ...back to orig #
     EAB5   DO B7         16825          bne  pezro       ; bad bits
     EAB7                 16826  pu30
     EAB7   F6 OO         16827          inc  $0,x        ; bump contents
     EAB9   C8            16828          iny
     EABA   DO FB         16829          bne  pu30        ; not done
                          16830
     EABC   D5 OO         16831          cmp  $0,x        ; check for good count
     EABE   DO AE         16832          bne  pezro       ; something's wrong
                          16833
     EACO   94 OO         16834          sty  $0,x        ; leave z-page zeroed
     EAC2   B5 OO         16835          lda  $0,x        ; check it
     EAC4   DO A8         16836          bne  pezro       ; wrong
                          16837
     EAC6   E8            16838          inx              ; next!
     EAC7   DO E9         16839          bne  pu20        ; not all done
                          16840
                          16841
                          16842  ; test 32k byte rom
                          16843
                          16844  ; enter x=start page
                          16845  ; exit if ok
                          16846
     EAC9   E6 6F         16847  rm10    inc  temp        ; next error #
     EACB   A2 80         16848          ldx  #128        ; 128 pages
     EACD   86 76         16849          stx  ip+1        ; save page, start x=0
     EACF   A9 OO         16850          lda  #0
     EAD1   85 75         16851          sta  ip          ; zero lo indirect
     EAD3   AO O2         16852          ldy  #2          ; skip signature bytes
     EAD5   18            16853          clc
     EAD6   E6 76         16854  rt10    inc  ip+1        ; do it backwards
     EAD8   71 75         16855  rt20    adc  (ip),y      ; total checksum in a
     EADA   C8            16856          iny
     EADB   DO FB         16857          bne  rt20
     EADD   CA            16858          dex
     EADE   DO F6         16859          bne  rt10
     EAEO   69 FF         16860          adc  #255         ; add in last carry
     EAE2   85 76         16861          sta  ip+1
     EAE4   DO 39         16862          bne  perr2       ; no - show error number
                          16863  ;       nop              ; /debug only
                          16864  ;       nop
```

```
Error Addr  Code           Seq    Source statement

                          16865
                          16866   ; **** rom ds 01/22/85 ***
                          16867
     EAE6   EA            16868           nop             ; fill
     EAE7   EA            16869           nop             ; fill
     EAE8   EA            16870           nop             ; fill
     EAE9   EA            16871           nop             ; fill
                          16872
                          16873   ;---------------------------------
                          16874
                          16875   ; test all common ram
                          16876
     EAEA   A9 01         16877   cr20    lda  #$01       ; start of 1st block
     EAEC   85 76         16878   cr30    sta  ip+1       ; save page #
     EAEE   E6 6F         16879           inc  temp       ; bump error #
                          16880
                          16881   ; enter x=# of pages in block
                          16882   ; ip ptr to first page in block
                          16883   ; exit if ok
                          16884
     EAF0   A2 07         16885   ramtst  ldx  #7         ; save page count
     EAF2   98            16886   ra10    tya             ; fill with adr sensitive pattern
     EAF3   18            16887           clc
     EAF4   65 76         16888           adc  ip+1
     EAF6   91 75         16889           sta  (ip),y
     EAF8   C8            16890           iny
     EAF9   D0 F7         16891           bne  ra10
     EAFB   E6 76         16892           inc  ip+1
     EAFD   CA            16893           dex
     EAFE   D0 F2         16894           bne  ra10
     EB00   A2 07         16895           ldx  #7         ; restore page count
     EB02   C6 76         16896   ra30    dec  ip+1       ; check pattern backwards
     EB04   88            16897   ra40    dey
     EB05   98            16898           tya             ; gen pattern again
     EB06   18            16899           clc
     EB07   65 76         16900           adc  ip+1
     EB09   D1 75         16901           cmp  (ip),y     ; ok ?
     EB0B   D0 12         16902           bne  perr2      ; no - show error #
     EB0D   49 FF         16903           eor  #$ff       ; yes - test inverse pattern
     EB0F   91 75         16904           sta  (ip),y
     EB11   51 75         16905           eor  (ip),y     ; ok ?
     EB13   91 75         16906           sta  (ip),y     ; leave memory zero
     EB15   D0 08         16907           bne  perr2      ; no - show error #
     EB17   98            16908           tya
     EB18   D0 EA         16909           bne  ra40
     EB1A   CA            16910           dex
     EB1B   D0 E5         16911           bne  ra30
                          16912
     EB1D   F0 03         16913           beq  diagok
                          16914
```

```
Error Addr  Code          Seq    Source statement

     EB1F   4C EA71      16915   perr2   jmp  perr
                         16916
     EB22                16917   diagok
                         16918   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         16919
     EB22   4C A7C0      16920           jmp  ptch31     ; *** rom ds 05/01/85 ***
                         16921   ;       ldx  #topwrt
                         16922   ;       txs
                         16923
                         16924   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         16925
     EB25   AD 1C00      16926   rtch31  lda  ledprt     ; clear leds
     EB28   29 F7        16927           and  #$ff-led0-led1
     EB2A   8D 1C00      16928           sta  ledprt
                         16929
     EB2D   A9 03        16930           lda  #3         ; neg edge of atn & edge on wp
     EB2F   8D 180C      16931           sta  pcr1
     EB32   A9 82        16932           lda  #%10000010 ; set,_t1,_t2,_cb1,_cb2,_sr,ca1,_ca2(wps)
     EB34   8D 180D      16933           sta  ifr1
     EB37   8D 180E      16934           sta  ier1
     EB3A   AD 1800      16935           lda  pb         ; compute primary addr
     EB3D   29 60        16936           and  #%01100000 ; pb5 and pb6 are unused lines
     EB3F   0A           16937           asl  a          ; shift to lower
     EB40   2A           16938           rol  a
     EB41   2A           16939           rol  a
     EB42   2A           16940           rol  a
     EB43   09 48        16941           ora  #$48       ; talk address
     EB45   85 78        16942           sta  tlkadr
     EB47   49 60        16943           eor  #$60       ; listen address
     EB49   85 77        16944           sta  lsnadr
                         16945
                         16946   ; initialize buffer pntr table
                         16947
     EB4B   A2 00        16948   inttab  ldx  #0
     EB4D   A0 00        16949           ldy  #0
     EB4F   A9 00        16950   intt1   lda  #0
     EB51   95 99        16951           sta  buftab,x
     EB53   E8           16952           inx
     EB54   B9 FEE0      16953           lda  bufind,y
     EB57   95 99        16954           sta  buftab,x
     EB59   E8           16955           inx
     EB5A   C8           16956           iny
     EB5B   C0 05        16957           cpy  #bfcnt
     EB5D   D0 F0        16958           bne  intt1
                         16959
     EB5F   A9 00        16960           lda  #<cmdbuf   ; set pntr to cmdbuf
     EB61   95 99        16961           sta  buftab,x
     EB63   E8           16962           inx
     EB64   A9 02        16963           lda  #>cmdbuf
     EB66   95 99        16964           sta  buftab,x
```

```
Error Addr  Code           Seq    Source statement

     EB68   E8           16965            inx
     EB69   A9 D5         16966            lda  #<errbuf    ; set pntr to errbuf
     EB6B   95 99         16967            sta  buftab,x
     EB6D   E8            16968            inx
     EB6E   A9 02         16969            lda  #>errbuf
     EB70   95 99         16970            sta  buftab,x
                          16971
     EB72   A9 FF         16972            lda  #$ff
     EB74   A2 12         16973            ldx  #maxsa
     EB76   9D 022B       16974   dskin1   sta  lintab,x
     EB79   CA            16975            dex
     EB7A   10 FA         16976            bpl  dskin1
                          16977
     EB7C   A2 05         16978            ldx  #mxchns-1
     EB7E                 16979   dskin2
     EB7E   95 A7         16980            sta  buf0,x      ; set buffers as unused
     EB80   95 AE         16981            sta  buf1,x
     EB82   95 CD         16982            sta  ss,x
     EB84   CA            16983            dex
     EB85   10 F7         16984            bpl  dskin2
                          16985
     EB87   A9 05         16986            lda  #bfcnt      ; set buffer pointers
     EB89   85 AB         16987            sta  buf0+cmdchn
     EB8B   A9 06         16988            lda  #bfcnt+1
     EB8D   85 AC         16989            sta  buf0+errchn
     EB8F   A9 FF         16990            lda  #$ff
     EB91   85 AD         16991            sta  buf0+blindx
     EB93   85 B4         16992            sta  buf1+blindx
                          16993
     EB95   A9 05         16994            lda  #errchn
     EB97   8D 023B       16995            sta  lintab+errsa
     EB9A   A9 84         16996            lda  #cmdchn+$80
     EB9C   8D 023A       16997            sta  lintab+cmdsa
     EB9F   A9 0F         16998            lda  #lxint      ; lindx 0 to 5 free
     EBA1   8D 0256       16999            sta  linuse
                          17000
     EBA4   A9 01         17001            lda  #rdylst
     EBA6   85 F6         17002            sta  chnrdy+cmdchn
     EBA8   A9 88         17003            lda  #rdytlk
     EBAA   85 F7         17004            sta  chnrdy+errchn
     EBAC   A9 E0         17005            lda  #$e0
     EBAE   8D 024F       17006            sta  bufuse
     EBB1   A9 FF         17007            lda  #$ff
     EBB3   8D 0250       17008            sta  bufuse+1
     EBB6   A9 01         17009            lda  #1
     EBB8   85 1C         17010            sta  wpsw
     EBBA   85 1D         17011            sta  wpsw+1
     EBBC   20 CB63       17012            jsr  usrint      ; init user jmp
     EBBF   20 CEFA       17013            jsr  lruint
                          17014
```

```
Error Addr  Code           Seq    Source statement

                          17015   ;**********************************
                          17016   ;
                          17017   ; controller initialization
                          17018   ;
                          17019   ;**********************************
                          17020
     EBC2   20 FF82       17021           jsr  ptch10     ; *** rom ds 05/01/85 controller init ***
                          17022   ;       jsr  cntint
                          17023
                          17024   ; set indirect vectors
                          17025
     EBC5   A9 22         17026           lda  #<diagok
     EBC7   85 65         17027           sta  vnmi
     EBC9   A9 EB         17028           lda  #>diagok
     EBCB   85 66         17029           sta  vnmi+1
                          17030
     EBCD   A9 06         17031           lda  #6         ; set up sector offset *** rom ds 01/22/85 ***
     EBCF   85 69         17032           sta  secinc
     EBD1   A9 05         17033           lda  #5
     EBD3   85 6A         17034           sta  revcnt     ; set up recovery count
                          17035
                          17036   ;*
                          17037   ;*******************************
                          17038   ;*
                          17039   ;*    seterr
                          17040   ;*    set up power on error msg
                          17041   ;*
                          17042   ;*******************************
                          17043   ;*
                          17044   ;*
     EBD5   A9 73         17045   seterr  lda  #$73
     EBD7   20 E6C1       17046           jsr  errts0
                          17047
                          17048
                          17049   ;must be contiguous to .file idle
                          17050
                          17051   ;*******************************
                          17052   ; init the serial bus
                          17053   ;
                          17054   ;*******************************
                          17055
                          17056   ;-------rom -05 8/18/83-----------------
     EBDA   A9 00         17057           lda  #$00       ;  data hi, clock hi,atna hi
     EBDC   8D 1800       17058           sta  pb
     EBDF   A9 1A         17059           lda  #%00011010 ;  atna,clkout,datout
     EBE1   8D 1802       17060           sta  ddrb1
                          17061   ;---------------------------------------
     EBE4   20 A786       17062           jsr  ptch29     ; *rom ds 02/01/85*
                          17063   ;       jsr  boot
                          17064           .include  idlesf
```

```
Error Addr  Code           Seq    Source statement

                          17066            .subttl 'idlesf.src'
                          17067
                          17068    ; idle loop, waiting for something to do
     EBE7                 17069    idle
     EBE7   58            17070            cli
                          17071
                          17072    ;  release all bus lines
                          17073
     EBE8   AD 1800       17074            lda  pb          ; clock and data high
     EBEB   29 E5         17075            and  #$e5        ; clock high
     EBED   8D 1800       17076            sta  pb          ; data high,atna hi
                          17077
     EBF0   AD 0255       17078            lda  cmdwat      ; test for pending command
     EBF3   F0 0A         17079            beq  idl1        ; no command waiting
                          17080
     EBF5   A9 00         17081            lda  #0
     EBF7   8D 0255       17082            sta  cmdwat
                          17083
     EBFA   EA            17084            nop              ; *** rom ds -06 *** / fill
     EBFB   EA            17085            nop              ; *** rom ds -06 *** / fill
                          17086    ;       sta  nmiflg      ; clear debounce
                          17087
                          17088    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17089
     EBFC   4C A61C       17090            jmp  ptch19      ; *rom ds 01/23/85*
                          17091    ;       jsr  parsxq      ; parse and xeq command
                          17092
     EBFF                 17093    rtch19                   ; return
                          17094
                          17095    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17096
     EBFF   58            17097    idl1     cli             ; test for drive running or openfile
     EC00   A5 7C         17098             lda  atnpnd
     EC02   F0 03         17099             beq  idl01
                          17100
                          17101    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17102
     EC04   4C A7B3       17103             jmp  ptch30      ; *rom ds 02/04/85*
                          17104    ;        jmp. atnsrv      ; service atn irq
                          17105
                          17106    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17107
     EC07                 17108    idl01
     EC07   58            17109             cli
     EC08   A9 0E         17110             lda  #14
     EC0A   85 72         17111             sta  temp+3
     EC0C   A9 00         17112             lda  #0          ; if file open, turn on act led
     EC0E   85 6F         17113             sta  temp
     EC10   85 70         17114             sta  temp+1
     EC12   A6 72         17115    idl2     ldx  temp+3      ; look thru lintab
```

```
Error Addr  Code           Seq    Source statement

     EC14   BD 022B      17116            lda  lintab,x    ; for active file
     EC17   C9 FF        17117            cmp  #$ff
     EC19   F0 10        17118            beq  idl3
                         17119
     EC1B   29 3F        17120            and  #$3f
     EC1D   85 82        17121            sta  lindx
     EC1F   20 DF93      17122            jsr  getact
     EC22   AA           17123            tax
     EC23   BD 025B      17124            lda  lstjob,x    ; determine which drv it is on
     EC26   29 01        17125            and  #1
     EC28   AA           17126            tax
     EC29   F6 6F        17127            inc  temp,x
     EC2B   C6 72        17128   idl3     dec  temp+3      ; set flag indicating drv
     EC2D   10 E3        17129            bpl  idl2        ; has file open
     EC2F   A0 04        17130            ldy  #bfcnt-1    ; look thru job que for
     EC31   B9 0000      17131   idl4     lda  jobs,y      ; for jobs still running
     EC34   10 05        17132            bpl  idl5
     EC36   29 01        17133            and  #1
     EC38   AA           17134            tax
     EC39   F6 6F        17135            inc  temp,x      ; set flag indicating drive
     EC3B   88           17136   idl5     dey              ; is active
     EC3C   10 F3        17137            bpl  idl4
     EC3E   78           17138            sei
                         17139
                         17140   ; dont allow irq when reading ledprt
                         17141
     EC3F   AD 1C00      17142            lda  ledprt
     EC42   29 F7        17143            and  #$ff-led0
     EC44   48           17144            pha
     EC45   A5 7F        17145            lda  drvnum
     EC47   85 86        17146            sta  r0
     EC49   A9 00        17147            lda  #0
     EC4B   85 7F        17148            sta  drvnum
     EC4D   A5 6F        17149            lda  temp
     EC4F   F0 0B        17150            beq  idl7
                         17151
     EC51   A5 1C        17152            lda  wpsw
     EC53   F0 03        17153            beq  idl6
                         17154
     EC55   20 D313      17155            jsr  cldchn
                         17156
     EC58                17157   idl6
     EC58   68           17158            pla              ; turn on led if drive flag
     EC59   09 08        17159            ora  #led0       ;  if not 0
     EC5B   48           17160            pha
     EC5C                17161   idl7
     EC5C   E6 7F        17162            inc  drvnum
     EC5E   A5 70        17163            lda  temp+1
     EC60   F0 0B        17164            beq  idl9
                         17165
```

```
Error Addr  Code           Seq    Source statement

     EC62   A5 1D        17166            lda  wpsw+1
     EC64   F0 03        17167            beq  idl8
                         17168
     EC66   20 D313      17169            jsr  cldchn
     EC69                17170  idl8
     EC69   68           17171            pla
     EC6A   09 00        17172            ora  #led1
     EC6C   48           17173            pha
     EC6D                17174  idl9
     EC6D   A5 86        17175            lda  r0
     EC6F   85 7F        17176            sta  drvnum
     EC71   68           17177            pla
     EC72   AE 026C      17178            ldx  erword
     EC75   F0 21        17179            beq  idl12      ; no error flashing
                         17180
     EC77   AD 1C00      17181            lda  ledprt     ; use current leds
     EC7A   E0 80        17182            cpx  #$80
     EC7C   D0 03        17183            bne  idl10      ; not 1st time
                         17184
     EC7E   4C EC8B      17185            jmp  idl11
                         17186
     EC81                17187  idl10
     EC81   AE 1805      17188            ldx  timer1
     EC84   30 12        17189            bmi  idl12      ; still timing
                         17190
     EC86   A2 A0        17191            ldx  #$a0       ; count 8 msec
     EC88   8E 1805      17192            stx  timer1
     EC8B                17193  idl11
     EC8B   CE 026C      17194            dec  erword     ; count units of 8 msec
     EC8E   D0 08        17195            bne  idl12      ; keep counting
                         17196
     EC90   4D 026D      17197            eor  erled      ; toggle led
     EC93   A2 10        17198            ldx  #16        ; count 16 units
     EC95   8E 026C      17199            stx  erword
     EC98                17200  idl12
     EC98   8D 1C00      17201            sta  ledprt     ; set leds
     EC9B   4C EBFF      17202            jmp  idl1       ; back to top of lop
                         17203            .include  lstdir
```

```
Error Addr  Code           Seq    Source statement

                          17205           .subttl 'lstdir.src'
                          17206
                          17207   ; start the directory loading function
                          17208   ; get the buffer and get it started
                          17209
     EC9E   A9 00         17210   stdir   lda  #0
     ECA0   85 83         17211           sta  sa
     ECA2   A9 01         17212           lda  #1          ; allocate chanl and 1 bufefer
     ECA4   20 D1E2       17213           jsr  getrch
                          17214
     ECA7   A9 00         17215           lda  #0
     ECA9   20 D4C8       17216           jsr  setpnt
                          17217
     ECAC   A6 82         17218           ldx  lindx
     ECAE   A9 00         17219           lda  #0
     ECB0   9D 0244       17220           sta  lstchr,x
     ECB3   20 DF93       17221           jsr  getact
     ECB6   AA            17222           tax
     ECB7   A5 7F         17223           lda  drvnum
     ECB9   9D 025B       17224           sta  lstjob,x
     ECBC   A9 01         17225           lda  #1          ;  put sal in buffer
     ECBE   20 CFF1       17226           jsr  putbyt
     ECC1   A9 04         17227           lda  #4          ; put sah in buffer
     ECC3   20 CFF1       17228           jsr  putbyt
     ECC6   A9 01         17229           lda  #1          ; insert fhoney links (0101)
     ECC8   20 CFF1       17230           jsr  putbyt
     ECCB   20 CFF1       17231           jsr  putbyt
     ECCE   AD 0272       17232           lda  nbtemp
     ECD1   20 CFF1       17233           jsr  putbyt      ; put in drvnum
     ECD4   A9 00         17234           lda  #0
     ECD6   20 CFF1       17235           jsr  putbyt
     ECD9   20 ED59       17236           jsr  movbuf      ; get disk name
     ECDC   20 DF93       17237           jsr  getact
     ECDF   0A            17238           asl  a
     ECE0   AA            17239           tax
     ECE1   D6 99         17240           dec  buftab,x
     ECE3   D6 99         17241           dec  buftab,x
     ECE5   A9 00         17242           lda  #0          ; end of this line
     ECE7   20 CFF1       17243           jsr  putbyt
     ECEA   A9 01         17244   dir1    lda  #1          ; insert fhoney links ($0101)
     ECEC   20 CFF1       17245           jsr  putbyt
     ECEF   20 CFF1       17246           jsr  putbyt
     ECF2   20 C6CE       17247           jsr  getnam      ; get #bufrs and file name
     ECF5   90 2C         17248           bcc  dir3        ; test if last entry
     ECF7   AD 0272       17249           lda  nbtemp
     ECFA   20 CFF1       17250           jsr  putbyt
     ECFD   AD 0273       17251           lda  nbtemp+1
     ED00   20 CFF1       17252           jsr  putbyt
     ED03   20 ED59       17253           jsr  movbuf
     ED06   A9 00         17254           lda  #0          ; end of entry
```

```
Error Addr  Code           Seq    Source statement

     ED08   20 CFF1      17255            jsr  putbyt
     ED0B   D0 DD        17256            bne  dir1
     ED0D   20 DF93      17257   dir10    jsr  getact
     ED10   0A           17258            asl  a
     ED11   AA           17259            tax
     ED12   A9 00        17260            lda  #0
     ED14   95 99        17261            sta  buftab,x
     ED16   A9 88        17262            lda  #rdytlk
     ED18   A4 82        17263            ldy  lindx
     ED1A   8D 0254      17264            sta  dirlst
     ED1D   99 00F2      17265            sta  chnrdy,y    ;  directory list buffer full
     ED20   A5 85        17266            lda  data
     ED22   60           17267            rts
                         17268
     ED23   AD 0272      17269   dir3     lda  nbtemp      ;  this is end of load
     ED26   20 CFF1      17270            jsr  putbyt
     ED29   AD 0273      17271            lda  nbtemp+1
     ED2C   20 CFF1      17272            jsr  putbyt
     ED2F   20 ED59      17273            jsr  movbuf
     ED32   20 DF93      17274            jsr  getact
     ED35   0A           17275            asl  a
     ED36   AA           17276            tax
     ED37   D6 99        17277            dec  buftab,x
     ED39   D6 99        17278            dec  buftab,x
     ED3B   A9 00        17279            lda  #0          ;  end of listing (000)
     ED3D   20 CFF1      17280            jsr  putbyt
     ED40   20 CFF1      17281            jsr  putbyt
     ED43   20 CFF1      17282            jsr  putbyt
     ED46   20 DF93      17283            jsr  getact
     ED49   0A           17284            asl  a
     ED4A   A8           17285            tay
     ED4B   B9 0099      17286            lda  buftab,y
     ED4E   A6 82        17287            ldx  lindx
     ED50   9D 0244      17288            sta  lstchr,x
     ED53   DE 0244      17289            dec  lstchr,x
     ED56   4C ED0D      17290            jmp  dir10
                         17291
                         17292   ; transfer file name to listing buffer
                         17293
     ED59   A0 00        17294   movbuf   ldy  #0
     ED5B   B9 02B1      17295   movb1    lda  nambuf,y
     ED5E   20 CFF1      17296            jsr  putbyt
     ED61   C8           17297            iny
     ED62   C0 1B        17298            cpy  #27
     ED64   D0 F5        17299            bne  movb1
     ED66   60           17300            rts
                         17301
                         17302   ; get char for directory loading
                         17303
     ED67   20 D137      17304   getdir   jsr  getbyt
```

```
Error Addr  Code           Seq    Source statement

     ED6A   F0 01        17305            beq  getd3
     ED6C   60           17306            rts
     ED6D   85 85        17307   getd3    sta  data
     ED6F   A4 82        17308            ldy  lindx
     ED71   B9 0244      17309            lda  lstchr,y
     ED74   F0 08        17310            beq  gd1
     ED76   A9 80        17311            lda  #eoiout
     ED78   99 00F2      17312            sta  chnrdy,y
     ED7B   A5 85        17313            lda  data
     ED7D   60           17314            rts
     ED7E                17315   gd1
     ED7E   48           17316            pha
     ED7F   20 ECEA      17317            jsr  dir1
     ED82   68           17318            pla
     ED83   60           17319            rts
                         17320            .include  verdir
```

```
Error Addr  Code           Seq    Source statement

                          17322            .subttl 'verdir.src'
                          17323    ;  validate files with bam
                          17324    ;  create new bam according to
                          17325    ;  contents of files entered in dir
                          17326
     ED84                 17327    verdir
     ED84                 17328    valdat
                          17329    ;validate is soft-load
     ED84    20 C1D1      17330             jsr  simprs      ; extract drive #
     ED87    20 D042      17331             jsr  initdr
     ED8A    A9 40        17332             lda  #$40
     ED8C    8D 02F9      17333             sta  wbam
                          17334
                          17335    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     EDCE   68           17372            pla
     EDCF   85 80        17373            sta  track
     EDD1   20 EDE5      17374            jsr  vmkbam      ; set bit used in bam
     EDD4   20 C604      17375   vd20     jsr  srre        ; search for more
     EDD7   F0 C3        17376            beq  vd10        ; no more files
     EDD9                17377   vd25
     EDD9   A0 00        17378            ldy  #0
     EDDB   B1 94        17379            lda  (dirbuf),y
     EDDD   30 D4        17380            bmi  vd15
     EDDF   20 C8B6      17381            jsr  deldir      ; not closed delete dir
     EDE2   4C EDD4      17382            jmp  vd20
                         17383
     EDE5                17384   vmkbam                    ; mark bam with file sectors
     EDE5   20 D55F      17385            jsr  tschk
     EDE8   20 EF90      17386            jsr  wused
     EDEB   20 D475      17387            jsr  opnird
     EDEE   A9 00        17388   mrk2     lda  #0
     EDF0   20 D4C8      17389            jsr  setpnt
     EDF3   20 D137      17390            jsr  getbyt
     EDF6   85 80        17391            sta  track
     EDF8   20 D137      17392            jsr  getbyt
     EDFB   85 81        17393            sta  sector
     EDFD   A5 80        17394            lda  track
     EDFF   D0 03        17395            bne  mrk1
     EE01   4C D227      17396            jmp  frechn
     EE04   20 EF90      17397   mrk1     jsr  wused
     EE07   20 D44D      17398            jsr  nxtbuf
     EE0A   4C EDEE      17399            jmp  mrk2
                         17400            .include  new
                         17401
```

```
Error Addr  Code           Seq    Source statement

                          17403           .subttl 'new.src'
                          17404
                          17405   ;  new: initialize a disk, disk is
                          17406   ;  soft-sectored, bit avail. map,
                          17407   ;  directory, & 1st block are all inited
                          17408
     EE0D   20 C312       17409   new     jsr  onedrv
     EE10   A5 E2         17410           lda  fildrv     ; set up drive #
     EE12   10 05         17411           bpl  n101
                          17412
     EE14   A9 33         17413           lda  #badfn     ; bad drive # given
     EE16   4C C1C8       17414           jmp  cmderr
                          17415
     EE19   29 01         17416   n101    and  #1
     EE1B   85 7F         17417           sta  drvnum
                          17418
                          17419   ;       jsr  setlds
     EE1D   20 FF9C       17420           jsr  ptch11     ; clr nodrv ***rom ds 01/21/85***
                          17421
     EE20   A5 7F         17422           lda  drvnum
     EE22   0A            17423           asl  a
     EE23   AA            17424           tax
     EE24   AC 027B       17425           ldy  filtbl+1   ; get disk id
     EE27   CC 0274       17426           cpy  cmdsiz     ; ?is this new or clear?
     EE2A   F0 1A         17427           beq  n108       ; end of cmd string
     EE2C   B9 0200       17428           lda  cmdbuf,y   ; format disk****
     EE2F   95 12         17429           sta  dskid,x    ; store in proper drive
     EE31   B9 0201       17430           lda  cmdbuf+1,y ; (y=0)
     EE34   95 13         17431           sta  dskid+1,x
                          17432
     EE36   20 D307       17433           jsr  clrchn     ; clear all channels when formatting
     EE39   A9 01         17434           lda  #1         ; ...in track, track=1
     EE3B   85 80         17435           sta  track
                          17436
                          17437   ;--------- patch7 for format bug 10/17/83---
                          17438   ;       jsr  format     ; transfer format to ram
     EE3D   20 FF2F       17439           jsr  patch7     ; set format flag
                          17440   ;----------------------------------------
                          17441
                          17442   ;<><><>
```

```
Error Addr  Code          Seq    Source statement

     EE46   20 D042      17453   n108    jsr  initdr     ; clear directory only
                         17454
     EE49   A6 7F        17455           ldx  drvnum
     EE4B   BD 0101      17456           lda  dskver,x   ; use current version #
     EE4E   CD FED5      17457           cmp  vernum
     EE51   F0 03        17458           beq  n110
     EE53   4C D572      17459           jmp  vnerr      ; wrong version #
                         17460
                         17461   ;<><><>
```

```
Error Addr  Code          Seq    Source statement

     EE96   20 EF93      17503            jsr  usedts      ; set bam block used
     EE99   A9 01        17504            lda  #1
     EE9B   85 81        17505            sta  sector
     EE9D   20 EF93      17506            jsr  usedts      ; set 1st dir block used
     EEA0   20 EEFF      17507            jsr  scrbam      ; scrub the bam
     EEA3   20 F005      17508            jsr  clrbam      ; set to all 0's
     EEA6   A0 01        17509            ldy  #1
     EEA8   A9 FF        17510            lda  #$ff        ; set end link
     EEAA   91 6D        17511            sta  (bmpnt),y
     EEAC   20 D464      17512            jsr  drtwrt      ; clear directory
     EEAF   C6 81        17513            dec  sector
                         17514
                         17515  ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          17525           .subttl 'map.src'
                          17526   ; build a new map on diskette
                          17527
     EEB7                 17528   newmap
     EEB7                 17529   newmpv
     EEB7   20 FOD1       17530           jsr  clnbam
     EEBA   AO OO         17531           ldy  #0
     EEBC   A9 12         17532           lda  #18        ; set link to 18.1
     EEBE   91 6D         17533           sta  (bmpnt),y
     EECO   C8            17534           iny
     EEC1   98            17535           tya
     EEC2   91 6D         17536           sta  (bmpnt),y
     EEC4   C8            17537           iny
     EEC5   C8            17538           iny
     EEC6   C8            17539           iny             ; .y=4
     EEC7                 17540   nm10
     EEC7   A9 OO         17541           lda  #0         ; clear track map
     EEC9   85 6F         17542           sta  t0
     EECB   85 7O         17543           sta  t1
     EECD   85 71         17544           sta  t2
                          17545
     EECF   98            17546           tya             ; 4=>1
     EEDO   4A            17547           lsr  a
     EED1   4A            17548           lsr  a          ; .a=track #
     EED2   2O F24B       17549           jsr  maxsec     ; store blks free byte away
     EED5   91 6D         17550           sta  (bmpnt),y
     EED7   C8            17551           iny
     EED8   AA            17552           tax
     EED9                 17553   nm2O
     EED9   38            17554           sec             ; set map bits
     EEDA   26 6F         17555           rol  t0         ;      t0          t1          t2
     EEDC   26 7O         17556           rol  t1         ;   76543210  111111      xxx21111
     EEDE   26 71         17557           rol  t2         ;             54321098        09876
     EEEO   CA            17558           dex             ;   11111111  11111111        11111
     EEE1   DO F6         17559           bne  nm2O
     EEE3                 17560   nm3O                    ; .x=0
     EEE3   B5 6F         17561           lda  t0,x
     EEE5   91 6D         17562           sta  (bmpnt),y  ; write out bit map
     EEE7   C8            17563           iny
     EEE8   E8            17564           inx
     EEE9   EO O3         17565           cpx  #3
     EEEB   9O F6         17566           bcc  nm3O
     EEED   CO 9O         17567           cpy  #$9O       ; end of bam, 4-143
     EEEF   9O D6         17568           bcc  nm10
                          17569
     EEF1   4C D075       17570           jmp  nfcalc     ; calc # free sectors
                          17571
                          17572   ; write out the bit map to
                          17573   ; the drive in lstjob(active)
                          17574
```

```
Error Addr  Code          Seq    Source statement

     EEF4   20 DF93      17575   mapout  jsr  getact
     EEF7   AA           17576           tax
     EEF8   BD 025B      17577           lda  lstjob,x
     EEFB   29 01        17578   mo10    and  #1
     EEFD   85 7F        17579           sta  drvnum     ; check bam before writing
                         17580
                         17581   ; write bam according to drvnum
                         17582
     EEFF                17583   scrbam
     EEFF   A4 7F        17584           ldy  drvnum
     EF01   B9 0251      17585           lda  mdirty,y
     EF04   D0 01        17586           bne  sb10
     EF06   60           17587           rts             ; not dirty
     EF07                17588   sb10
     EF07   A9 00        17589           lda  #0         ; set to clean bam
     EF09   99 0251      17590           sta  mdirty,y
     EF0C   20 EF3A      17591           jsr  setbpt     ; set bit map ptr
     EF0F   A5 7F        17592           lda  drvnum
     EF11   0A           17593           asl  a
     EF12   48           17594           pha
                         17595   ;put memory images to bam
     EF13   20 F0A5      17596           jsr  putbam
     EF16   68           17597           pla
     EF17   18           17598           clc
     EF18   69 01        17599           adc  #1
     EF1A   20 F0A5      17600           jsr  putbam
                         17601   ; verify the bam block count
                         17602   ; matches the bits
                         17603
     EF1D   A5 80        17604           lda  track
     EF1F   48           17605           pha             ; save track var
     EF20   A9 01        17606           lda  #1
     EF22   85 80        17607           sta  track
     EF24                17608   sb20
     EF24   0A           17609           asl  a
     EF25   0A           17610           asl  a
     EF26   85 6D        17611           sta  bmpnt
                         17612
                         17613   ;<><><>
```

```
Error Addr  Code          Seq   Source statement

     EF35   85 80       17625           sta  track
                        17626
                        17627   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                        17628
     EF37   4C A58D     17629           jmp  wrt_bam
                        17630   ;       jmp  dowrit     ; write it out
                        17631
                        17632   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

```
Error Addr  Code           Seq    Source statement

                          17634
                          17635   ; set bit map ptr, read in bam if nec.
                          17636
     EF3A                 17637   setbpt
     EF3A   20 F1OF       17638           jsr  bam2a
     EF3D   AA            17639           tax
     EF3E   20 FODF       17640           jsr  redbam      ; read bam if not in
     EF41   A6 F9         17641           ldx  jobnum
     EF43   BD FEEO       17642           lda  bufind,x    ; set the ptr
     EF46   85 6E         17643           sta  bmpnt+1
     EF48   A9 OO         17644           lda  #0
     EF4A   85 6D         17645           sta  bmpnt
     EF4C   6O            17646           rts
                          17647
                          17648   ; calc the number of free blocks on drvnum
                          17649
     EF4D                 17650   numfre
     EF4D   A6 7F         17651           ldx  drvnum
     EF4F   BD O2FA       17652           lda  ndbl,x
     EF52   8D O272       17653           sta  nbtemp
     EF55   BD O2FC       17654           lda  ndbh,x
     EF58   8D O273       17655           sta  nbtemp+1
     EF5B   6O            17656           rts
                          17657           .include  frets
```

```
Error Addr  Code           Seq    Source statement

                          17659            .subttl 'frets.src'
                          17660
                          17661    ; mark a track,sector as free in bam
     EF5C                 17662    wfree
     EF5C   20 EFF1       17663            jsr  fixbam
                          17664
     EF5F                 17665    frets
                          17666    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17667
     EF5F   4C A83E       17668            jmp  ptch33      ; *** rom 02/22/85 ***
                          17669    ;       jsr  freuse      ; calc index into bam
                          17670
                          17671    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17672
     EF62                 17673    frets2
     EF62   38            17674            sec              ; flag for no action
     EF63   D0 22         17675            bne  frerts      ; free already
     EF65   B1 6D         17676            lda  (bmpnt),y   ; not free, free it
     EF67   1D EFE9       17677            ora  bmask,x
     EF6A   91 6D         17678            sta  (bmpnt),y
     EF6C   20 EF88       17679            jsr  dtybam      ; set dirty flag
     EF6F   A4 6F         17680            ldy  temp
     EF71   18            17681            clc
     EF72   B1 6D         17682            lda  (bmpnt),y   ; add one
     EF74   69 01         17683            adc  #1
     EF76   91 6D         17684            sta  (bmpnt),y
     EF78   A5 80         17685            lda  track
     EF7A   CD FE85       17686            cmp  dirtrk
     EF7D   F0 3B         17687            beq  use10
                          17688
     EF7F   FE 02FA       17689    fre20   inc  ndbl,x
     EF82   D0 03         17690            bne  fre10
     EF84   FE 02FC       17691            inc  ndbh,x
     EF87                 17692    fre10
     EF87   60            17693    frerts  rts
                          17694
                          17695
                          17696
     EF88                 17697    dtybam
     EF88   A6 7F         17698            ldx  drvnum
     EF8A   A9 01         17699            lda  #1
     EF8C   9D 0251       17700            sta  mdirty,x    ; set dirty flag
     EF8F   60            17701            rts
                          17702
                          17703
                          17704    ;
                          17705    ; mark track,sector,(bmpnt) as used
                          17706    ;
     EF90                 17707    wused
     EF90   20 EFF1       17708            jsr  fixbam
```

```
Error Addr  Code           Seq    Source statement

                          17709
      EF93                17710  usedts                   ; calc in dex into bam
                          17711  ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17712
      EF93   4C A874      17713          jmp  ptch34       ; *** rom 02/22/85 ***
                          17714  ;       jsr  freuse       ; calc index into bam
                          17715
                          17716  ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          17717
      EF96   F0 36        17718  rtch34  beq  userts       ; used, no action
      EF98   B1 6D        17719          lda  (bmpnt),y    ; get bits
      EF9A   5D EFE9      17720          eor  bmask,x      ; mark sec used
      EF9D   91 6D        17721          sta  (bmpnt),y
      EF9F   20 EF88      17722          jsr  dtybam
      EFA2   A4 6F        17723          ldy  temp
      EFA4   B1 6D        17724          lda  (bmpnt),y    ; get count
      EFA6   38           17725          sec
      EFA7   E9 01        17726          sbc  #1           ; dec one (c=0)
      EFA9   91 6D        17727          sta  (bmpnt),y    ; save it
      EFAB   A5 80        17728          lda  track
      EFAD   CD FE85      17729          cmp  dirtrk
      EFB0   F0 0B        17730          beq  use20
                          17731
      EFB2   BD 02FA      17732  use30   lda  ndbl,x
      EFB5   D0 03        17733          bne  use10
      EFB7   DE 02FC      17734          dec  ndbh,x
      EFBA                17735  use10
      EFBA   DE 02FA      17736          dec  ndbl,x
      EFBD                17737  use20
      EFBD   BD 02FC      17738          lda  ndbh,x
      EFC0   D0 0C        17739          bne  userts
      EFC2   BD 02FA      17740          lda  ndbl,x
      EFC5   C9 03        17741          cmp  #3
      EFC7   B0 05        17742          bcs  userts
                          17743
      EFC9   A9 72        17744          lda  #dskful
      EFCB   20 E6C7      17745          jsr  errmsg
      EFCE   60           17746  userts  rts
                          17747
                          17748  ; calculates index into bam
                          17749  ; for frets and usedts
                          17750
      EFCF                17751  freuse
      EFCF   20 F011      17752          jsr  setbam
      EFD2   98           17753          tya
      EFD3                17754  freus2
      EFD3   85 6F        17755          sta  temp         ; save index
      EFD5                17756  freus3
      EFD5   A5 81        17757          lda  sector       ; a=sector/8
      EFD7   4A           17758          lsr  a
```

```
Error Addr  Code           Seq    Source statement

     EFD8   4A           17759            lsr  a
     EFD9   4A           17760            lsr  a           ; for which of three bytes
     EFDA   38           17761            sec
     EFDB   65 6F        17762            adc  temp        ; calc index
     EFDD   A8           17763            tay
     EFDE   A5 81        17764            lda  sector      ; bit in that byte
     EFE0   29 07        17765            and  #7
     EFE2   AA           17766            tax
     EFE3   B1 6D        17767            lda  (bmpnt),y   ; get the byte
     EFE5   3D EFE9      17768            and  bmask,x     ; test it
     EFE8   60           17769            rts              ; z=1=used,z=0=free
                         17770
                         17771
     EFE9   01 02 04     17772   bmask     .byte  1,2,4,8,16,32,64,128
     EFEC   08 10 20
     EFEF   40 80
                         17773
     EFF1                17774   fixbam                     ; write the bam according to wbam flag
     EFF1   A9 FF        17775            lda  #$ff
     EFF3   2C 02F9      17776            bit  wbam
     EFF6   F0 0C        17777            beq  fbam10      ; test flags
     EFF8   10 0A        17778            bpl  fbam10
     EFFA   70 08        17779            bvs  fbam10
                         17780
     EFFC   A9 00        17781            lda  #0
     EFFE   8D 02F9      17782            sta  wbam        ; clear flag
                         17783
                         17784   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     F00B               17807   clb1
     F00B   91 6D       17808           sta  (bmpnt),y
     F00D   C8          17809           iny
     F00E   D0 FB       17810           bne  clb1
     F010   60          17811           rts
                        17812
     F011               17813   setbam                  ; set bam image in memory
     F011   A5 6F       17814           lda  t0         ; save temps
     F013   48          17815           pha
     F014   A5 70       17816           lda  t1
     F016   48          17817           pha
                        17818
                        17819   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     F048   69 A1        17857            adc  #<bam       ; set actual ptr
     F04A   85 6D        17858            sta  bmpnt
     F04C   A9 02        17859            lda  #>bam
     F04E   69 00        17860            adc  #0
     F050   85 6E        17861            sta  bmpnt+1
     F052   A0 00        17862            ldy  #0
     F054   68           17863            pla
     F055   85 70        17864            sta  t1
     F057   68           17865            pla
     F058   85 6F        17866            sta  t0
     F05A   60           17867            rts
                         17868
                         17869
     F05B                17870  swap                      ; swap images of bam
     F05B   A6 6F        17871            ldx  t0
     F05D   20 F0DF      17872            jsr  redbam      ; read bam if not in
     F060   A5 7F        17873            lda  drvnum
     F062   AA           17874            tax
     F063   0A           17875            asl  a
     F064   1D 029B      17876            ora  ubam,x      ; swap out least used image
     F067   49 01        17877            eor  #1
     F069   29 03        17878            and  #3
     F06B   85 70        17879            sta  t1
     F06D   20 F0A5      17880            jsr  putbam      ; put to bam
     F070   A5 F9        17881            lda  jobnum
     F072   0A           17882            asl  a
     F073   AA           17883            tax
     F074   A5 80        17884            lda  track
     F076   0A           17885            asl  a
     F077   0A           17886            asl  a
     F078   95 99        17887            sta  buftab,x    ; set ptr
     F07A   A5 70        17888            lda  t1
     F07C   0A           17889            asl  a
     F07D   0A           17890            asl  a
     F07E   A8           17891            tay
     F07F                17892  swap3                     ; transfer bam to mem image
     F07F   A1 99        17893            lda  (buftab,x)
     F081   99 02A1      17894            sta  bam,y
     F084   A9 00        17895            lda  #0
     F086   81 99        17896            sta  (buftab,x) ; clear bam
     F088   F6 99        17897            inc  buftab,x
     F08A   C8           17898            iny
     F08B   98           17899            tya
     F08C   29 03        17900            and  #3
     F08E   D0 EF        17901            bne  swap3
                         17902
     F090   A6 70        17903            ldx  t1
     F092   A5 80        17904            lda  track
     F094   9D 029D      17905            sta  tbam,x      ; set track # for image
                         17906
```

```
Error Addr  Code           Seq    Source statement

     F097   AD 02F9      17907            lda  wbam
     F09A   D0 03        17908            bne  swap4       ; don't write now
                         17909
                         17910  ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         17911
     F09C   4C A58D      17912            jmp  wrt_bam     ; *** rom ds 02/22/85
                         17913  ;         jmp  dowrit
                         17914
                         17915  ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                         17916
     F09F                17917  swap4
     F09F   09 80        17918            ora  #$80        ; set pending write flag
     F0A1   8D 02F9      17919            sta  wbam
     F0A4   60           17920            rts
                         17921
     F0A5                17922  putbam                     ; put mem image to bam
     F0A5   A8           17923            tay
     F0A6   B9 029D      17924            lda  tbam,y
     F0A9   F0 25        17925            beq  swap2       ; no image here
                         17926
     F0AB   48           17927            pha              ; save track #
     F0AC   A9 00        17928            lda  #0
     F0AE   99 029D      17929            sta  tbam,y      ; clear track flag
     F0B1   A5 F9        17930            lda  jobnum
     F0B3   0A           17931            asl  a
     F0B4   AA           17932            tax
     F0B5   68           17933            pla
     F0B6   0A           17934            asl  a
     F0B7   0A           17935            asl  a
     F0B8   95 99        17936            sta  buftab,x    ; set ptr in bam
     F0BA   98           17937            tya
     F0BB   0A           17938            asl  a
     F0BC   0A           17939            asl  a
     F0BD   A8           17940            tay
     F0BE                17941  swap1                      ; transfer image to bam
     F0BE   B9 02A1      17942            lda  bam,y
     F0C1   81 99        17943            sta  (buftab,x)
     F0C3   A9 00        17944            lda  #0
     F0C5   99 02A1      17945            sta  bam,y       ; clear image
     F0C8   F6 99        17946            inc  buftab,x
     F0CA   C8           17947            iny
     F0CB   98           17948            tya
     F0CC   29 03        17949            and  #3
     F0CE   D0 EE        17950            bne  swap1
     F0D0                17951  swap2
     F0D0   60           17952            rts
                         17953
                         17954
     F0D1                17955  clnbam                     ; clean track # for images
     F0D1   A5 7F        17956            lda  drvnum
```

```
Error Addr  Code          Seq    Source statement

     FOD3   OA           17957            asl  a
     FOD4   AA           17958            tax
     FOD5   A9 00        17959            lda  #0
     FOD7   9D 029D      17960            sta  tbam,x
     FODA   E8           17961            inx
     FODB   9D 029D      17962            sta  tbam,x
     FODE   60           17963            rts
                         17964
                         17965
                         17966
     FODF                17967   redbam                    ; read in bam if not present
     FODF   B5 A7        17968            lda  buf0,x
     FOE1   C9 FF        17969            cmp  #$ff
     FOE3   D0 25        17970            bne  rbm20        ; it is in memory
     FOE5   8A           17971            txa
     FOE6   48           17972            pha               ; save channel ptr
     FOE7   20 D28E      17973            jsr  getbuf       ; go find a buffer
     FOEA   AA           17974            tax
     FOEB   10 05        17975            bpl  rbm10
                         17976
     FOED   A9 70        17977            lda  #nochnl      ; no buffers around
     FOEF   20 C1C8      17978            jsr  cmderr
     FOF2                17979   rbm10
     FOF2   86 F9        17980            stx  jobnum       ; save jobnum assigned
     FOF4   68           17981            pla
     FOF5   A8           17982            tay
     FOF6   8A           17983            txa
     FOF7   09 80        17984            ora  #$80
     FOF9   99 00A7      17985            sta  buf0,y       ; set as inactive for stealing
                         17986
                         17987   ;read in bam
                         17988
     FOFC   0A           17989            asl  a
     FOFD   AA           17990            tax
     FOFE   AD FE85      17991            lda  dirtrk
     F101   95 06        17992            sta  hdrs,x
     F103   A9 00        17993            lda  #0
     F105   95 07        17994            sta  hdrs+1,x
                         17995
                         17996   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                         18007   ;set bam pointer in buf0/1 tables
                         18008
     F10F                18009   bam2a                   ; leave in .a
     F10F   A9 06        18010           lda  #blindx
     F111   A6 7F        18011           ldx  drvnum
     F113   D0 03        18012           bne  b2x10
                         18013
     F115   18           18014           clc
     F116   69 07        18015           adc  #mxchns+1
     F118                18016   b2x10
     F118   60           18017           rts
                         18018
     F119                18019   bam2x                   ; leave in .x
     F119   20 F10F      18020           jsr  bam2a
     F11C   AA           18021           tax
     F11D   60           18022           rts
                         18023           .include  tstfnd
```

```
Error Addr  Code          Seq    Source statement

                         18025           .subttl 'tstfnd.src'
                         18026
                         18027   ; next track & sector
                         18028   ;  returns next available track & sector
                         18029   ;  given    current  t  &  s
                         18030
                         18031   ;  allocation is from track 18
                         18032   ;  towards 1 & 35, by full tracks
     F11E                18033   nxtts
     F11E   20 DE3E      18034           jsr  gethdr
     F121   A9 03        18035           lda  #3
     F123   85 6F        18036           sta  temp
     F125   A9 01        18037           lda  #1         ; set no write bam
     F127   0D 02F9      18038           ora  wbam
     F12A   8D 02F9      18039           sta  wbam
     F12D                18040   nxtds
     F12D                18041   nxt1
                         18042   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

     F15A   A9 72        18075   nxterr  lda  #dskful
     F15C   20 C1C8      18076           jsr  cmderr
     F15F   C6 80        18077   nxt2    dec  track
     F161   D0 CA        18078           bne  nxt1
                         18079
     F163   AE FE85      18080           ldx  dirtrk
     F166   E8           18081           inx
     F167   86 80        18082           stx  track
     F169   A9 00        18083           lda  #0
     F16B   85 81        18084           sta  sector
     F16D   C6 6F        18085           dec  temp
     F16F   D0 BC        18086           bne  nxt1
                         18087
     F171   F0 E7        18088           beq  nxterr
                         18089
                         18090   ; find the next optimum sector
                         18091   ; next sector=current sector+n
                         18092
     F173   A5 81        18093   fndnxt  lda  sector
     F175   18           18094           clc
     F176   65 69        18095           adc  secinc
     F178   85 81        18096           sta  sector
     F17A   A5 80        18097           lda  track
     F17C   20 F24B      18098           jsr  maxsec
     F17F   8D 024E      18099           sta  lstsec
     F182   8D 024D      18100           sta  cmd
     F185   C5 81        18101           cmp  sector
     F187   B0 0C        18102           bcs  fndn0
                         18103
     F189   38           18104           sec
     F18A   A5 81        18105           lda  sector
     F18C   ED 024E      18106           sbc  lstsec
     F18F   85 81        18107           sta  sector
     F191   F0 02        18108           beq  fndn0
                         18109
     F193   C6 81        18110           dec  sector
     F195                18111   fndn0
     F195   20 F1FA      18112           jsr  getsec
     F198   F0 03        18113           beq  fndn2
     F19A                18114   fndn1
     F19A   4C EF90      18115           jmp  wused
     F19D                18116   fndn2
     F19D   A9 00        18117           lda  #0
     F19F   85 81        18118           sta  sector
     F1A1   20 F1FA      18119           jsr  getsec
     F1A4   D0 F4        18120           bne  fndn1
                         18121
     F1A6   4C F1F5      18122           jmp  derr
                         18123
                         18124
```

```
Error Addr  Code           Seq    Source statement

                          18125   ; returns optimum initial track,sector
                          18126
     F1A9                 18127   intts
     F1A9   A9 01         18128           lda  #1
     F1AB   0D 02F9       18129           ora  wbam
     F1AE   8D 02F9       18130           sta  wbam
     F1B1   A5 86         18131           lda  r0
     F1B3   48            18132           pha                 ; save temp var
                          18133                               ;r0:= 1
     F1B4   A9 01         18134           lda  #1
     F1B6   85 86         18135           sta  r0
     F1B8                 18136   its1                        ; track:= dirtrk-r0
     F1B8   AD FE85       18137           lda  dirtrk
     F1BB   38            18138           sec
     F1BC   E5 86         18139           sbc  r0
     F1BE   85 80         18140           sta  track
                          18141                               ;if t>0
     F1C0   90 09         18142           bcc  its2
                          18143
     F1C2   F0 07         18144           beq  its2
                          18145                               ;then begin
                          18146
                          18147   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          18148
     F1C4   4C A905       18149           jmp  ptch37         ; *** rom ds 02/22/85 ***
                          18150   ;       jsr  setbam         ; set the bam pointer
                          18151
                          18152   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          18153
                          18154                               ;if @b .y  then goto fndsec
     F1C7   B1 6D         18155   rtch37  lda  (bmpnt),y
     F1C9   D0 1B         18156   rtch37a bne  fndsec
                          18157                               ;end
     F1CB                 18158   its2                        ; track:= dirtrk+r0
     F1CB   AD FE85       18159           lda  dirtrk
     F1CE   18            18160           clc
     F1CF   65 86         18161           adc  r0
     F1D1   85 80         18162           sta  track
                          18163                               ;r0:= r0+1
     F1D3   E6 86         18164           inc  r0
                          18165                               ;if track >=maxtrk then cmder2(systs)
     F1D5   CD 02AC       18166           cmp  maxtrk
     F1D8   90 05         18167           bcc  its3
                          18168
     F1DA   A9 67         18169           lda  #systs
     F1DC   20 E645       18170           jsr  cmder2
     F1DF                 18171   its3
                          18172   ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          18173
     F1DF   4C A91E       18174           jmp  ptch38         ; *** rom ds 02/22/85 ***
```

```
Error Addr  Code           Seq    Source statement

                          18175   ;       jsr  setbam      ; set the bam pointer
                          18176
                          18177   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          18225
     F219   E6 81         18226            inc  sector
     F21B   D0 F0         18227            bne  gs10        ; bra
     F21D                 18228   gs20
     F21D   A9 00         18229            lda  #0
     F21F                 18230   gs30
     F21F   60            18231            rts              ; (z=1): used
                          18232
                          18233   ;bit map validity check
     F220                 18234   avck
     F220   A5 6F         18235            lda  temp
     F222   48            18236            pha              ; save temp
     F223   A9 00         18237            lda  #0
     F225   85 6F         18238            sta  temp        ; temp:=0
                          18239                             ;for .y:=bamsiz to 1 do;
     F227   AC FE86       18240            ldy  bamsiz
     F22A   88            18241            dey
     F22B                 18242   ac10                      ; for .x:=7 to 0 do;
     F22B   A2 07         18243            ldx  #7          ; count the bits
     F22D                 18244   ac20                      ; if @b .y  & bmask x
                          18245                             ;  then temp:=temp+1
     F22D   B1 6D         18246            lda  (bmpnt),y
     F22F   3D EFE9       18247            and  bmask,x
     F232   F0 02         18248            beq  ac30
     F234   E6 6F         18249            inc  temp
     F236                 18250   ac30                      ; end .x
     F236   CA            18251            dex
     F237   10 F4         18252            bpl  ac20
                          18253                             ;end .y
     F239   88            18254            dey
     F23A   D0 EF         18255            bne  ac10
                          18256                             ;if @b .y  <> temp
                          18257                             ;  then cmder2(direrr);
     F23C   B1 6D         18258            lda  (bmpnt),y
     F23E   C5 6F         18259            cmp  temp
     F240   D0 04         18260            bne  ac40        ; counts do not match
                          18261
     F242   68            18262            pla
     F243   85 6F         18263            sta  temp        ; restore temp
     F245   60            18264            rts
     F246                 18265   ac40
     F246   A9 71         18266            lda  #direrr
     F248   20 E645       18267            jsr  cmder2
                          18268
                          18269   ; .a=track # ,returns #sectors on this track
                          18270
     F24B                 18271   maxsec
                          18272   ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                         18275    ;       ldx  nzones
                         18276
                         18277    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          18300   ;
            =F259         18301   cntst=  *
                          18302   ;
                          18303   ;
                          18304   ;
                          18305   ;     defs for low cost controller
                          18306   ;
                          18307   ;
                          18308   ;     written by glenn stark
                          18309   ;     4/1/80
                          18310   ;
                          18311   ;
                          18312   ;   (c) commodore business machines
                          18313   ;
            =1805         18314   timer1   =$1805          ;  timer 1 counter
                          18315   ;
                          18316   ;
                          18317   ;
                          18318   ;   mos 6522
                          18319   ;   address $1c00
                          18320   ;
            =1C00         18321           *  =$1c00
                          18322   ;
      1C00  =1C01         18323   dskcnt   *=*+1           ;  port b
                          18324   ; disk i/o control lines
                          18325   ; bit 0: step in
                          18326   ; bit 1: step out
                          18327   ; bit 2: -motor on
                          18328   ; bit 3: act led
                          18329   ; bit 4: write protect sense
                          18330   ; bit 5: density select 0
                          18331   ; bit 6: density select 1
                          18332   ; bit 7: sync detect
                          18333   ;
                          18334   ;
      1C01  =1C02         18335   data2    *=*+1           ;  port a
                          18336   ; gcr data input and output port
                          18337   ;
      1C02  =1C03         18338   ddrb2    *=*+1           ;  data direction control
      1C03  =1C04         18339   ddra2    *=*+1           ;  data direction control
                          18340   ;
      1C04  =1C05         18341   t1lc2    *=*+1           ;  timer 1 low counter
      1C05  =1C06         18342   t1hc2    *=*+1           ;  timer 1 hi countr
                          18343   ;
      1C06  =1C07         18344   t1ll2    *=*+1           ;  timer 1 low latch
      1C07  =1C08         18345   t1hl2    *=*+1           ;  timer 1 hi latch
                          18346   ;
      1C08  =1C09         18347   t2ll2    *=*+1           ;  timer two low latch
      1C09  =1C0A         18348   t2lh2    *=*+1           ;  timer two hi latch
                          18349   ;
```

```
Error Addr  Code           Seq    Source statement

     1C0A    =1C0B        18350   sr2      *=*+1          ;  shift register
                          18351   ;
     1C0B    =1C0C        18352   acr2     *=*+1
                          18353   ;
     1C0C    =1C0D        18354   pcr2     *=*+1
                          18355   ;
     1C0D    =1C0E        18356   ifr2     *=*+1
                          18357   ;
     1C0E    =1C0F        18358   ier2     *=*+1
                          18359   ;
                          18360   ;
             =F259        18361           *  =cntst
                          18362   ;
                          18363   ;
                          18364   ;
                          18365   ;.end
                          18366           .include  lccinit
                          18367           .subttl 'lccinit.src'
```

```
Error Addr  Code           Seq    Source statement

                          18369
                          18370  ;      initialization of controller
                          18371
     F259   A9 6F         18372  cntint  lda  #%01101111          ;  data direction
     F25B   8D 1C02       18373          sta  ddrb2
     F25E   29 F0         18374          and  #$ff-$08-$04-$03    ;  turn motor off,set phase a, led off
     F260   8D 1C00       18375          sta  dskcnt
                          18376
                          18377  ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          18419
     F29C   A9 05         18420           lda  #<inact
     F29E   85 62         18421           sta  nxtst
     F2A0   A9 FA         18422           lda  #>inact
     F2A2   85 63         18423           sta  nxtst+1
                          18424
     F2A4   A9 C8         18425           lda  #200
     F2A6   85 64         18426           sta  minstp
                          18427
     F2A8   A9 04         18428           lda  #4
     F2AA   85 5E         18429           sta  as
                          18430
     F2AC   A9 04         18431           lda  #$4
     F2AE   85 5F         18432           sta  af
                          18433           .include  lcccntrl
```

```
Error Addr  Code           Seq    Source statement

                         18435            .subttl  'lcccntrl.src'
                         18436    ;
                         18437    ;
                         18438    ;
                         18439    ;     *contrl
                         18440    ;
                         18441    ;     main controller loop
                         18442    ;
                         18443    ;     scans job que for jobs
                         18444    ;
                         18445    ;    finds job on current track
                         18446    ;    if it exists
                         18447    ;
     F2B0                18448    lcc
                         18449    ;
     F2B0   BA           18450            tsx              ;  save current stack pointer
     F2B1   86 49        18451            stx  savsp
                         18452    ;
     F2B3   AD 1C04      18453            lda  t1lc2       ; reset irq flag
                         18454    ;
     F2B6   AD 1C0C      18455            lda  pcr2        ;  enable s.o. to 6502
     F2B9   09 0E        18456            ora  #$0e        ;  hi output
     F2BB   8D 1C0C      18457            sta  pcr2
                         18458    ;
                         18459    ;
                         18460    ;
     F2BE   A0 05        18461    top     ldy  #numjob-1   ;  pointer into job que
                         18462    ;
     F2C0                18463    cont10
     F2C0   B9 0000      18464            lda  jobs,y      ;  find a job (msb set)
     F2C3   10 2E        18465            bpl  cont20      ;  not one here
                         18466
     F2C5   C9 D0        18467            cmp  #jumpc      ;  test if its a jump command
     F2C7   D0 04        18468            bne  cont30
                         18469    ;
     F2C9   98           18470            tya              ;  put job num in .a
     F2CA   4C F370      18471            jmp  ex2
                         18472    ;
                         18473    ;
     F2CD                18474    cont30
     F2CD   29 01        18475            and  #1          ;  get drive #
     F2CF   F0 07        18476            beq  cont35
                         18477    ;
     F2D1   84 3F        18478            sty  jobn
     F2D3   A9 0F        18479            lda  #$0f        ; bad drive # error
     F2D5   4C F969      18480            jmp  errr
                         18481    ;
     F2D8   AA           18482    cont35  tax
     F2D9   85. 3D       18483            sta  drive
                         18484    ;
```

```
Error Addr  Code           Seq    Source statement

     F2DB   C5 3E        18485            cmp  cdrive      ;  test if current drive
     F2DD   F0 0A        18486            beq  cont40
                         18487    ;
     F2DF   20 F97E      18488            jsr  turnon      ;  turn on drive
     F2E2   A5 3D        18489            lda  drive
     F2E4   85 3E        18490            sta  cdrive
     F2E6   4C F99C      18491            jmp  end         ;  go clean up
                         18492    ;
                         18493    ;
     F2E9   A5 20        18494    cont40  lda  drvst       ;  test if motor up to speed
     F2EB   30 03        18495            bmi  cont50
                         18496    ;
     F2ED   0A           18497            asl  a           ;  test if stepping
     F2EE   10 09        18498            bpl  que         ;  not stepping
                         18499    ;
     F2F0   4C F99C      18500    cont50  jmp  end
                         18501    ;
     F2F3   88           18502    cont20  dey
     F2F4   10 CA        18503            bpl  cont10
                         18504    ;
     F2F6   4C F99C      18505            jmp  end
                         18506    ;
                         18507    ;
     F2F9   A9 20        18508    que     lda  #$20        ;  status=running
     F2FB   85 20        18509            sta  drvst
                         18510    ;
     F2FD   A0 05        18511            ldy  #numjob-1
     F2FF   84 3F        18512            sty  jobn
                         18513    ;
     F301   20 F393      18514    que10   jsr  setjb
     F304   30 1A        18515            bmi  que20
                         18516    ;
     F306   C6 3F        18517    que05   dec  jobn
     F308   10 F7        18518            bpl  que10
                         18519    ;
                         18520    ;
     F30A   A4 41        18521            ldy  nxtjob
     F30C   20 F395      18522            jsr  setjb1
                         18523    ;
     F30F   A5 42        18524            lda  nxtrk
     F311   85 4A        18525            sta  steps
     F313   06 4A        18526            asl  steps       ;  steps*2
                         18527    ;
     F315   A9 60        18528            lda  #$60        ;  set status=stepping
     F317   85 20        18529            sta  drvst
                         18530    ;
                         18531    ;
     F319   B1 32        18532            lda  (hdrpnt),y          ;  get dest track #
     F31B   85 22        18533            sta  drvtrk
     F31D   4C F99C      18534    fin     jmp  end
```

```
Error Addr  Code           Seq    Source statement

                          18535   ;
                          18536   ;
     F320   29 01         18537   que20   and  #1          ;  test if same drive
     F322   C5 3D         18538           cmp  drive
     F324   D0 E0         18539           bne  que05
                          18540   ;
     F326   A5 22         18541           lda  drvtrk
     F328   F0 12         18542           beq  gotu        ;  uninit. track #
                          18543   ;
     F32A   38            18544           sec              ;  calc distance to track
     F32B   F1 32         18545           sbc  (hdrpnt),y
     F32D   F0 0D         18546           beq  gotu        ;  on track
                          18547   ;
     F32F   49 FF         18548           eor  #$ff        ;  2's comp
     F331   85 42         18549           sta  nxtrk
     F333   E6 42         18550           inc  nxtrk
                          18551   ;
     F335   A5 3F         18552           lda  jobn        ;  save job# and dist to track
     F337   85 41         18553           sta  nxtjob
                          18554   ;
     F339   4C F306       18555           jmp  que05
                          18556   ;
                          18557   ;
                          18558   ;
                          18559   ;
     F33C   A2 04         18560   gotu    ldx  #4          ;  set track and sectr
     F33E   B1 32         18561           lda  (hdrpnt),y
     F340   85 40         18562           sta  tracc
                          18563   ;
     F342   DD FED6       18564   gotu10  cmp  trknum-1,x
                          18565   ;
     F345   CA            18566           dex
     F346   B0 FA         18567           bcs  gotu10
                          18568   ;
     F348   BD FED1       18569           lda  numsec,x
     F34B   85 43         18570           sta  sectr
                          18571   ;
     F34D   8A            18572           txa              ;  set density
     F34E   0A            18573           asl  a
     F34F   0A            18574           asl  a
     F350   0A            18575           asl  a
     F351   0A            18576           asl  a
     F352   0A            18577           asl  a
     F353   85 44         18578           sta  work
                          18579   ;
     F355   AD 1C00       18580           lda  dskcnt
     F358   29 9F         18581           and  #$9f        ;  clear density bits
     F35A   05 44         18582           ora  work
     F35C   8D 1C00       18583           sta  dskcnt
                          18584   ;
```

```
Error Addr  Code           Seq    Source statement

     F35F   A6 3D        18585            ldx  drive       ;  drive num in .x
                         18586   ;
     F361   A5 45        18587            lda  job         ;  yes, go do the job
     F363   C9 40        18588            cmp  #bumpc      ;  test for bump
     F365   F0 15        18589            beq  bmp
                         18590   ;
                         18591   ;
     F367   C9 60        18592   exe      cmp  #execd      ;  test if execute
     F369   F0 03        18593            beq  ex
                         18594   ;
     F36B   4C F3B1      18595            jmp  seak        ;  do a sector seek
                         18596   ;
     F36E   A5 3F        18597   ex       lda  jobn        ;  jump to buffer
     F370   18           18598   ex2      clc
     F371   69 03        18599            adc  #>bufs
     F373   85 31        18600            sta  bufpnt+1
     F375   A9 00        18601            lda  #0
     F377   85 30        18602            sta  bufpnt
     F379   6C 0030      18603   ex3      jmp  (bufpnt)
                         18604   ;
                         18605   ;
     F37C                18606   bmp
     F37C   A9 60        18607            lda  #$60        ;  set status=stepping
     F37E   85 20        18608            sta  drvst
                         18609   ;
     F380   AD 1C00      18610            lda  dskcnt
     F383   29 FC        18611            and  #$ff-$03    ;  set phase a
     F385   8D 1C00      18612            sta  dskcnt
                         18613   ;
                         18614   ;
     F388   A9 A4        18615            lda  #256-92     ;  step back 45 traks
     F38A   85 4A        18616            sta  steps
                         18617   ;
     F38C   A9 01        18618            lda  #1          ;  drvtrk now 1
     F38E   85 22        18619            sta  drvtrk
                         18620   ;
     F390   4C F969      18621            jmp  errr        ;  job done return 1
                         18622   ;
                         18623   ;
     F393   A4 3F        18624   setjb    ldy  jobn
     F395   B9 0000      18625   setjb1   lda  jobs,y
     F398   48           18626            pha
     F399   10 10        18627            bpl  setj10      ;  no job here
                         18628   ;
     F39B   29 78        18629            and  #$78
     F39D   85 45        18630            sta  job
     F39F   98           18631            tya
     F3A0   0A           18632            asl  a
     F3A1   69 06        18633            adc  #<hdrs
     F3A3   85 32        18634            sta  hdrpnt
```

```
Error Addr  Code           Seq    Source statement

     F3A5   98           18635            tya              ;  point at buffer
     F3A6   18           18636            clc
     F3A7   69 03        18637            adc  #>bufs
     F3A9   85 31        18638            sta  bufpnt+1
                         18639    ;
                         18640    ;
     F3AB   A0 00        18641    setj10  ldy  #0
     F3AD   84 30        18642            sty  bufpnt
                         18643    ;
     F3AF   68           18644            pla
     F3B0   60           18645            rts
                         18646    ;
                         18647    ;
                         18648    ;
                         18649    ;.end
                         18650            .include  lccseek
```

```
Error Addr  Code           Seq    Source statement

                          18652            .subttl 'lccseek.src'
                          18653   ;
                          18654   ;
                          18655   ;
     F3B1   A2 5A         18656   seak     ldx  #90          ;  search 90 headers
     F3B3   86 4B         18657            stx  tmp
                          18658   ;
     F3B5   A2 00         18659            ldx  #0           ; read in 8 gcr bytes
                          18660   ;
     F3B7   A9 52         18661            lda  #$52         ;  header block id
     F3B9   85 24         18662            sta  stab
                          18663   ;
     F3BB   20 F556       18664   seek10   jsr  sync        ;  find sync char
                          18665   ;
     F3BE   50 FE         18666            bvc  *            ;  wait for block id
     F3C0   B8            18667            clv
                          18668   ;
     F3C1   AD 1C01       18669            lda  data2
     F3C4   C5 24         18670            cmp  stab         ;  test if header block
     F3C6   D0 3F         18671            bne  seek20       ;  not header
                          18672   ;
     F3C8   50 FE         18673   seek15   bvc  *
     F3CA   B8            18674            clv              ;  read in gcr header
                          18675   ;
     F3CB   AD 1C01       18676            lda  data2
     F3CE   95 25         18677            sta  stab+1,x
                          18678   ;
     F3D0   E8            18679            inx
     F3D1   E0 07         18680            cpx  #7
     F3D3   D0 F3         18681            bne  seek15
                          18682   ;
     F3D5   20 F497       18683            jsr  cnvbin       ;  convert header in stab to binary in header
                          18684   ;
     F3D8   A0 04         18685            ldy  #4           ;  compute checksum
     F3DA   A9 00         18686            lda  #0
                          18687   ;
     F3DC   59 0016       18688   seek30   eor  header,y
     F3DF   88            18689            dey
     F3E0   10 FA         18690            bpl  seek30
                          18691   ;
     F3E2   C9 00         18692            cmp  #0           ;  test if ok
     F3E4   D0 38         18693            bne  cserr        ;  nope, checksum error in header
                          18694   ;
     F3E6   A6 3E         18695            ldx  cdrive       ;  update drive track#
     F3E8   A5 18         18696            lda  header+2
     F3EA   95 22         18697            sta  drvtrk,x
                          18698   ;
     F3EC   A5 45         18699            lda  job          ;  test if a seek job
     F3EE   C9 30         18700            cmp  #$30
     F3F0   F0 1E         18701            beq  eseek
```

```
Error Addr  Code           Seq    Source statement

                          18702   ;
     F3F2   A5 3E         18703           lda  cdrive
     F3F4   0A            18704           asl  a          ;  test if correct id
     F3F5   A8            18705           tay
     F3F6   B9 0012       18706           lda  dskid,y
     F3F9   C5 16         18707           cmp  header
     F3FB   D0 1E         18708           bne  badid
     F3FD   B9 0013       18709           lda  dskid+1,y
     F400   C5 17         18710           cmp  header+1
     F402   D0 17         18711           bne  badid
                          18712   ;
     F404   4C F423       18713           jmp  wsect      ;  find best sector to service
                          18714   ;
                          18715   ;
     F407   C6 4B         18716   seek20  dec  tmp        ;  search more?
     F409   D0 B0         18717           bne  seek10     ; yes
                          18718   ;
     F40B   A9 02         18719           lda  #2         ;  cant find a sector
     F40D   20 F969       18720           jsr  errr
                          18721   ;
                          18722   ;
     F410   A5 16         18723   eseek   lda  header     ; harris fix....
     F412   85 12         18724           sta  dskid      ; ....
     F414   A5 17         18725           lda  header+1   ; ....
     F416   85 13         18726           sta  dskid+1    ; ....
                          18727   ;
     F418   A9 01         18728   done    lda  #1         ;  return ok code
     F41A   2C            18729           .byte    skip2
                          18730   ;
     F41B   A9 0B         18731   badid   lda  #11        ;  disk id mismatch
     F41D   2C            18732           .byte    skip2
                          18733   ;
     F41E   A9 09         18734   cserr   lda  #9         ;  checksum error in header
     F420   4C F969       18735           jmp  errr
                          18736   ;
                          18737   ;
                          18738   ;
     F423   A9 7F         18739   wsect   lda  #$7f       ;  find best job
     F425   85 4C         18740           sta  csect
                          18741   ;
     F427   A5 19         18742           lda  header+3   ;  get upcoming sector #
     F429   18            18743           clc
     F42A   69 02         18744           adc  #2
     F42C   C5 43         18745           cmp  sectr
     F42E   90 02         18746           bcc  1460
                          18747   ;
     F430   E5 43         18748           sbc  sectr      ; wrap around
                          18749   ;
     F432   85 4D         18750   1460    sta  nexts      ;  next sector
                          18751   ;
```

```
Error Addr  Code           Seq    Source statement

     F434   A2 05        18752            ldx  #numjob-1
     F436   86 3F        18753            stx  jobn
                         18754  ;
     F438   A2 FF        18755            ldx  #$ff
                         18756  ;
     F43A   20 F393      18757  1480      jsr  setjb
     F43D   10 44        18758            bpl  1470
                         18759  ;
     F43F   85 44        18760            sta  work
     F441   29 01        18761            and  #1
     F443   C5 3E        18762            cmp  cdrive      ;  test if same drive
     F445   D0 3C        18763            bne  1470        ;  nope
                         18764  ;
     F447   A0 00        18765            ldy  #0          ;  test if same track
     F449   B1 32        18766            lda  (hdrpnt),y
     F44B   C5 40        18767            cmp  tracc
     F44D   D0 34        18768            bne  1470
                         18769  ;
     F44F   A5 45        18770            lda  job         ;  test if execute job
     F451   C9 60        18771            cmp  #execd
     F453   F0 0C        18772            beq  1465
                         18773  ;
     F455   A0 01        18774            ldy  #1
     F457   38           18775            sec
     F458   B1 32        18776            lda  (hdrpnt),y
     F45A   E5 4D        18777            sbc  nexts
     F45C   10 03        18778            bpl  1465
                         18779  ;
     F45E   18           18780            clc
     F45F   65 43        18781            adc  sectr
                         18782  ;
     F461   C5 4C        18783  1465      cmp  csect
     F463   B0 1E        18784            bcs  1470
                         18785  ;
     F465   48           18786            pha              ;  save it
     F466   A5 45        18787            lda  job
     F468   F0 14        18788            beq  tstrdj      ;  must be a read
                         18789  ;
     F46A   68           18790            pla
     F46B   C9 09        18791            cmp  #wrtmin     ;    if(csect<9)return;
     F46D   90 14        18792            bcc  1470        ;    if(csect>12)return;
                         18793  ;
     F46F   C9 0C        18794            cmp  #wrtmax
     F471   B0 10        18795            bcs  1470
                         18796  ;
     F473   85 4C        18797  doitt     sta  csect       ; its better
     F475   A5 3F        18798            lda  jobn
     F477   AA           18799            tax
     F478   69 03        18800            adc  #>bufs
     F47A   85 31        18801            sta  bufpnt+1
```

```
Error Addr  Code           Seq    Source statement

                          18802   ;
     F47C   D0 05         18803           bne  1470
                          18804   ;
     F47E   68            18805   tstrdj  pla
     F47F   C9 06         18806           cmp  #rdmax     ;  if(csect>6)return;
     F481   90 F0         18807           bcc  doitt
                          18808   ;
                          18809   ;
     F483   C6 3F         18810   1470    dec  jobn
     F485   10 B3         18811           bpl  1480
                          18812   ;
     F487   8A            18813           txa             ;  test if a job to do
     F488   10 03         18814           bpl  1490
                          18815   ;
     F48A   4C F99C       18816           jmp  end        ;  no job found
                          18817   ;
     F48D   86 3F         18818   1490    stx  jobn
     F48F   20 F393       18819           jsr  setjb
     F492   A5 45         18820           lda  job
     F494   4C F4CA       18821           jmp  reed
                          18822   ;
                          18823   ;
                          18824   ;
                          18825   ;
     F497   A5 30         18826   cnvbin  lda  bufpnt
     F499   48            18827           pha
     F49A   A5 31         18828           lda  bufpnt+1
     F49C   48            18829           pha             ;  save buffer pntr
                          18830   ;
     F49D   A9 24         18831           lda  #<stab
     F49F   85 30         18832           sta  bufpnt     ;  point at gcr code
     F4A1   A9 00         18833           lda  #>stab
     F4A3   85 31         18834           sta  bufpnt+1
                          18835   ;
     F4A5   A9 00         18836           lda  #0
     F4A7   85 34         18837           sta  gcrpnt
                          18838   ;
     F4A9   20 F7E6       18839           jsr  get4gb     ;  convert 4 bytes
                          18840   ;
     F4AC   A5 55         18841           lda  btab+3
     F4AE   85 18         18842           sta  header+2
                          18843   ;
     F4B0   A5 54         18844           lda  btab+2
     F4B2   85 19         18845           sta  header+3
                          18846   ;
     F4B4   A5 53         18847           lda  btab+1
     F4B6   85 1A         18848           sta  header+4
                          18849   ;
                          18850   ;
     F4B8   20 F7E6       18851           jsr  get4gb     ;  get 2 more
```

```
Error Addr  Code           Seq    Source statement

                          18852   ;
     F4BB   A5 52         18853           lda  btab       ;  get id
     F4BD   85 17         18854           sta  header+1
     F4BF   A5 53         18855           lda  btab+1
     F4C1   85 16         18856           sta  header
                          18857   ;
     F4C3   68            18858           pla
     F4C4   85 31         18859           sta  bufpnt+1   ; restore pointer
     F4C6   68            18860           pla
     F4C7   85 30         18861           sta  bufpnt
                          18862   ;
     F4C9   60            18863           rts
                          18864   ;
                          18865   ;
                          18866   ;
                          18867   ;
                          18868   ;
                          18869   ;
                          18870   ;
                          18871   ;.end      .
                          18872           .include  lccread
```

```
Error Addr  Code           Seq    Source statement

                          18874            .subttl  'lcc.read'
                          18875   ;
                          18876   ;
                          18877   ;
                          18878   ;   *read
                          18879   ;
                          18880   ;    read in track,sector specified
                          18881   ;    in header
                          18882   ;
                          18883   ;
     F4CA   C9 00         18884   reed    cmp  #0          ;  test if read job
     F4CC   F0 03         18885           beq  read01      ;  go test if write
     F4CE   4C F56E       18886           jmp  wright
                          18887   ;
     F4D1   20 F50A       18888   read01  jsr  dstrt       ;  find header and start reading data
                          18889   ;
     F4D4   50 FE         18890   read11  bvc  *           ;  wait for byte
     F4D6   B8            18891           clv
                          18892   ;
     F4D7   AD 1C01       18893           lda  data2       ;  store away data
     F4DA   91 30         18894           sta  (bufpnt),y  ;  in data buffer
     F4DC   C8            18895           iny
     F4DD   D0 F5         18896           bne  read11
                          18897   ;
     F4DF   A0 BA         18898           ldy  #255-toprd  ;  store rest in overflow buffer
                          18899   ;
     F4E1   50 FE         18900   read20  bvc  *
     F4E3   B8            18901           clv
                          18902   ;
     F4E4   AD 1C01       18903           lda  data2
     F4E7   99 0100       18904           sta  ovrbuf,y
     F4EA   C8            18905           iny
     F4EB   D0 F4         18906           bne  read20
                          18907   ;
     F4ED   20 F8E0       18908           jsr  gcrbin      ;  convert buffer to binary
                          18909   ;
     F4F0   A5 38         18910           lda  bid         ;  test if its a data block
     F4F2   C5 47         18911           cmp  dbid
     F4F4   F0 05         18912           beq  read28
                          18913   ;
     F4F6   A9 04         18914           lda  #4          ;  not a data block
     F4F8   4C F969       18915           jmp  errr
                          18916   ;
     F4FB   20 F5E9       18917   read28  jsr  chkblk     ;  calc checksum
                          18918   ;
     F4FE   C5 3A         18919           cmp  chksum
     F500   F0 03         18920           beq  read40
                          18921   ;
     F502   A9 05         18922           lda  #5          ;  data block checksum error
     F504   2C            18923           .byte  skip2
```

```
Error Addr  Code           Seq    Source statement

                          18924  ;
     F505   A9 01          18925  read40  lda  #1          ;  read data block ok
     F507   4C F969        18926          jmp  errr
                          18927  ;
                          18928  ;
                          18929  ;
     F50A   20 F510        18930  dstrt   jsr  srch        ;  find header
     F50D   4C F556        18931          jmp  sync        ;  and then data block sync
                          18932  ;
                          18933  ;
     F510   A5 3D          18934  srch    lda  drive       ;  create header image
     F512   0A             18935          asl  a
     F513   AA             18936          tax
                          18937  ;
     F514   B5 12          18938          lda  dskid,x     ;  get master id for the drive
     F516   85 16          18939          sta  header
     F518   B5 13          18940          lda  dskid+1,x
     F51A   85 17          18941          sta  header+1
                          18942
     F51C   A0 00          18943          ldy  #0          ;  get track,sector
     F51E   B1 32          18944          lda  (hdrpnt),y
     F520   85 18          18945          sta  header+2
     F522   C8             18946          iny
     F523   B1 32          18947          lda  (hdrpnt),y
     F525   85 19          18948          sta  header+3
                          18949  ;
     F527   A9 00          18950          lda  #0
                          18951
                          18952  ;create header checksum
                          18953
     F529   45 16          18954          eor  header
     F52B   45 17          18955          eor  header+1
     F52D   45 18          18956          eor  header+2
     F52F   45 19          18957          eor  header+3
                          18958  ;
     F531   85 1A          18959          sta  header+4    ;  store the checksum
                          18960  ;
     F533   20 F934        18961          jsr  conhdr      ;  convert header to gcr
                          18962  ;
     F536   A2 5A          18963          ldx  #90         ;  search 90 sync chars
                          18964  ;
     F538   20 F556        18965  srch20  jsr  sync       ;  find sync
                          18966  ;
     F53B   A0 00          18967          ldy  #0          ;  test 8 gcr bytes
                          18968  ;
     F53D   50 FE          18969  srch25  bvc  *
     F53F   B8             18970          clv              ;  wait for byte
                          18971  ;
     F540   AD 1C01        18972          lda  data2
     F543   D9 0024        18973          cmp  stab,y      ;  test if the same
```

```
Error Addr  Code           Seq    Source statement

     F546   D0 06        18974            bne   srch30      ;  nope
                         18975    ;
     F548   C8           18976            iny
     F549   C0 08        18977            cpy   #8
     F54B   D0 F0        18978            bne   srch25
                         18979    ;
     F54D   60           18980            rts
                         18981    ;
                         18982    ;
     F54E   CA           18983    srch30  dex                 ; try again
     F54F   D0 E7        18984            bne   srch20
                         18985    ;
     F551   A9 02        18986            lda   #2            ;  cant find this header
     F553   4C F969      18987    err     jmp   errr
                         18988    ;
                         18989    ;
     F556                18990    sync
                         18991    ;
     F556   A9 D0        18992            lda   #$80+80       ;  wait 20 ms for sync max
     F558   8D 1805      18993            sta   timer1
                         18994    ;
     F55B   A9 03        18995            lda   #3            ;  error code for no sync
                         18996    ;
     F55D   2C 1805      18997    sync10  bit   timer1        ;  test for time out
     F560   10 F1        18998            bpl   err
     F562   2C 1C00      18999            bit   dskcnt        ;  test for sync
     F565   30 F6        19000            bmi   sync10
                         19001    ;
                         19002    ;
     F567   AD 1C01      19003            lda   data2         ;  reset pa latch
     F56A   B8           19004            clv
     F56B   A0 00        19005            ldy   #0            ;  clear pointer
     F56D   60           19006            rts
                         19007    ;
                         19008    ;
                         19009    ;
                         19010    ;.end
                         19011            .include  lccwrt
```

```
Error Addr  Code           Seq    Source statement

                          19013            .subttl 'lccwrt.src'
                          19014    ;
                          19015    ;
                          19016    ;
                          19017    ;     * write job
                          19018    ;
                          19019    ;     write out data buffer
                          19020    ;
                          19021    ;
     F56E   C9 10         19022    wright  cmp  #$10        ;  test if write
     F570   F0 03         19023            beq  wrt05
                          19024    ;
     F572   4C F691       19025            jmp  vrfy
                          19026    ;
     F575   20 F5E9       19027    wrt05   jsr  chkblk      ;  get block checksum
     F578   85 3A         19028            sta  chksum
                          19029
     F57A   AD 1C00       19030            lda  dskcnt      ;  test for write protect
     F57D   29 10         19031            and  #$10
     F57F   D0 05         19032            bne  wrt10       ;  not  protected
                          19033    ;
     F581   A9 08         19034            lda  #8          ;  write protect error
     F583   4C F969       19035            jmp  errr
                          19036    ;
     F586   20 F78F       19037    wrt10   jsr  bingcr      ;  convert buffer to write image
                          19038    ;
     F589   20 F510       19039            jsr  srch        ;  find header
                          19040    ;
     F58C   A2 09         19041            ldx  #gap1-2     ;  wait out header gap
                          19042    ;
     F58E   50 FE         19043    wrt20   bvc  *
     F590   B8            19044            clv
                          19045    ;
     F591   CA            19046            dex              ;  test if done yet
     F592   D0 FA         19047            bne  wrt20
                          19048    ;
     F594   A9 FF         19049            lda  #$ff        ;  make output $ff
     F596   8D 1C03       19050            sta  ddra2
                          19051    ;
     F599   AD 1C0C       19052            lda  pcr2        ;  set write mode
     F59C   29 1F         19053            and  #$ff-$e0
     F59E   09 C0         19054            ora  #$c0
     F5A0   8D 1C0C       19055            sta  pcr2
                          19056    ;
     F5A3   A9 FF         19057            lda  #$ff        ;  write 5 gcr sync
     F5A5   A2 05         19058            ldx  #numsyn     ;
     F5A7   8D 1C01       19059            sta  data2
     F5AA   B8            19060            clv
                          19061    ;
     F5AB   50 FE         19062    wrtsnc  bvc  *
```

```
Error Addr  Code           Seq    Source statement

                          19063   ;
      F5AD   B8           19064           clv
      F5AE   CA           19065           dex
      F5AF   DO FA        19066           bne  wrtsnc
                          19067   ;
                          19068   ;  write out overflow buffer
                          19069   ;
      F5B1   AO BB        19070           ldy  #256-topwrt
                          19071   ;
      F5B3   B9 0100      19072   wrt30   lda  ovrbuf,y   ;  get a char
      F5B6   50 FE        19073           bvc  *           ;  wait until ready
      F5B8   B8           19074           clv
                          19075   ;
      F5B9   8D 1C01      19076           sta  data2       ;  stuff it
      F5BC   C8           19077           iny
      F5BD   DO F4        19078           bne  wrt30       ;  do next char
                          19079   ;
                          19080   ; write rest of buffer
                          19081   ;
      F5BF   B1 30        19082   wrt40   lda  (bufpnt),y ;  now do buffer
      F5C1   50 FE        19083           bvc  *           ;  wait until ready
      F5C3   B8           19084           clv
                          19085   ;
      F5C4   8D 1C01      19086           sta  data2       ;  stuff it again
      F5C7   C8           19087           iny
                          19088   ;
                          19089   ; test if done
                          19090   ;
      F5C8   DO F5        19091           bne  wrt40       ;  do the whole thing
                          19092   ;
      F5CA   50 FE        19093           bvc  *           ;  wait for last char to write out
                          19094   ;
                          19095   ;
      F5CC   AD 1C0C      19096           lda  pcr2        ;  goto read mode
      F5CF   09 E0        19097           ora  #$e0
      F5D1   8D 1C0C      19098           sta  pcr2
                          19099   ;
      F5D4   A9 00        19100           lda  #0          ;  make data2 input $00
      F5D6   8D 1C03      19101           sta  ddra2
                          19102   ;
      F5D9   20 F5F2      19103           jsr  wtobin      ;  convert write image to binary
                          19104   ;
      F5DC   A4 3F        19105           ldy  jobn        ;  make job a verify
      F5DE   B9 0000      19106           lda  jobs,y
      F5E1   49 30        19107           eor  #$30
      F5E3   99 0000      19108           sta  jobs,y
                          19109   ;
      F5E6   4C F3B1      19110           jmp  seak        ;  scan job que
                          19111   ;
                          19112   ;
```

```
Error Addr  Code           Seq    Source statement

     F5E9   A9 00        19113   chkblk  lda  #0          ;  get eor checksum
     F5EB   A8           19114           tay
                         19115   ;
     F5EC   51 30        19116   chkb10  eor  (bufpnt),y
     F5EE   C8           19117           iny
     F5EF   D0 FB        19118           bne  chkb10
                         19119   ;
     F5F1   60           19120           rts             ;  return checksum in .a
                         19121   ;
                         19122   ;
                         19123   ;
                         19124   ;
                         19125   ;    * wtobin
                         19126   ;
                         19127   ;    convert write image back to
                         19128   ;    binary data
                         19129   ;
                         19130   ;
     F5F2   A9 00        19131   wtobin  lda  #0          ;  init pointer
     F5F4   85 2E        19132           sta  savpnt
     F5F6   85 30        19133           sta  bufpnt      ;  lsb
     F5F8   85 4F        19134           sta  nxtpnt
                         19135   ;
     F5FA   A5 31        19136           lda  bufpnt+1    ; goto buffer next
     F5FC   85 4E        19137           sta  nxtbf
                         19138   ;
     F5FE   A9 01        19139           lda  #>ovrbuf    ;  use overflow first
     F600   85 31        19140           sta  bufpnt+1
     F602   85 2F        19141           sta  savpnt+1
                         19142   ;
     F604   A9 BB        19143           lda  #256-topwrt
     F606   85 34        19144           sta  gcrpnt      ;  get here first
     F608   85 36        19145           sta  bytcnt      ;  store here
                         19146   ;
     F60A   20 F7E6      19147           jsr  get4gb      ;  get first four- id and 3 data
                         19148   ;
     F60D   A5 52        19149           lda  btab        ;  save bid
     F60F   85 38        19150           sta  bid
                         19151   ;
     F611   A4 36        19152           ldy  bytcnt
                         19153   ;
     F613   A5 53        19154           lda  btab+1
     F615   91 2E        19155           sta  (savpnt),y
     F617   C8           19156           iny
                         19157   ;
     F618   A5 54        19158           lda  btab+2
     F61A   91 2E        19159           sta  (savpnt),y
     F61C   C8           19160           iny
                         19161   ;
     F61D   A5 55        19162           lda  btab+3
```

```
Error Addr  Code           Seq    Source statement

     F61F   91 2E        19163            sta  (savpnt),y
     F621   C8           19164            iny
                         19165  ;
     F622   84 36        19166            sty  bytcnt
                         19167  ;
     F624   20 F7E6      19168  wtob14    jsr  get4gb      ; do rest of overflow buffer
                         19169  ;
     F627   A4 36        19170            ldy  bytcnt
                         19171  ;
     F629   A5 52        19172            lda  btab
     F62B   91 2E        19173            sta  (savpnt),y
     F62D   C8           19174            iny
                         19175  ;
     F62E   A5 53        19176            lda  btab+1
     F630   91 2E        19177            sta  (savpnt),y
     F632   C8           19178            iny
     F633   F0 0E        19179            beq  wtob50
                         19180  ;
     F635   A5 54        19181            lda  btab+2
     F637   91 2E        19182            sta  (savpnt),y
     F639   C8           19183            iny
                         19184  ;
     F63A   A5 55        19185            lda  btab+3
     F63C   91 2E        19186            sta  (savpnt),y
     F63E   C8           19187            iny
                         19188  ;
     F63F   84 36        19189            sty  bytcnt
     F641   D0 E1        19190            bne  wtob14      ;  jmp
                         19191  ;
     F643                19192  wtob50
                         19193  ;
     F643   A5 54        19194            lda  btab+2
     F645   91 30        19195            sta  (bufpnt),y
     F647   C8           19196            iny
                         19197  ;
     F648   A5 55        19198            lda  btab+3
     F64A   91 30        19199            sta  (bufpnt),y
     F64C   C8           19200            iny
                         19201  ;
     F64D   84 36        19202            sty  bytcnt
                         19203  ;
     F64F   20 F7E6      19204  wtob53    jsr  get4gb
                         19205  ;
     F652   A4 36        19206            ldy  bytcnt
                         19207  ;
     F654   A5 52        19208            lda  btab
     F656   91 30        19209            sta  (bufpnt),y
     F658   C8           19210            iny
                         19211  ;
     F659   A5 53        19212            lda  btab+1
```

```
Error Addr  Code           Seq    Source statement

     F65B   91 30        19213            sta  (bufpnt),y
     F65D   C8           19214            iny
                         19215   ;
     F65E   A5 54        19216            lda  btab+2
     F660   91 30        19217            sta  (bufpnt),y
     F662   C8           19218            iny
                         19219   ;
     F663   A5 55        19220            lda  btab+3
     F665   91 30        19221            sta  (bufpnt),y
     F667   C8           19222            iny
                         19223   ;
     F668   84 36        19224            sty  bytcnt
     F66A   C0 BB        19225            cpy  #187
     F66C   90 E1        19226            bcc  wtob53
                         19227   ;
     F66E   A9 45        19228   wtob52   lda  #69        ;  move buffer up
     F670   85 2E        19229            sta  savpnt
                         19230   ;
     F672   A5 31        19231            lda  bufpnt+1
     F674   85 2F        19232            sta  savpnt+1
                         19233   ;
     F676   A0 BA        19234            ldy  #256-topwrt-1
                         19235   ;
     F678   B1 30        19236   wtob55   lda  (bufpnt),y
     F67A   91 2E        19237            sta  (savpnt),y
                         19238   ;
     F67C   88           19239            dey
     F67D   D0 F9        19240            bne  wtob55
                         19241   ;
     F67F   B1 30        19242            lda  (bufpnt),y
     F681   91 2E        19243            sta  (savpnt),y
                         19244   ;
                         19245   ;  move overflow over to buffer
                         19246   ;
     F683   A2 BB        19247            ldx  #256-topwrt
                         19248   ;
     F685   BD 0100      19249   wtob57   lda  ovrbuf,x
     F688   91 30        19250            sta  (bufpnt),y
                         19251   ;
     F68A   C8           19252            iny
     F68B   E8           19253            inx
     F68C   D0 F7        19254            bne  wtob57
                         19255   ;
     F68E   86 50        19256            stx  gcrflg     ;  clear buffer gcr flag
                         19257   ;
                         19258   ;
     F690   60           19259            rts
                         19260   ;
                         19261   ;
                         19262   ;
```

```
Error Addr  Code           Seq    Source statement

                          19263   ;
                          19264   ;
                          19265   ;     * verify data block
                          19266   ;
                          19267   ;   convert to gcr verify image
                          19268   ;
                          19269   ;   test against data block
                          19270   ;
                          19271   ;   convert back to binary
                          19272   ;
                          19273   ;
     F691   C9 20         19274   vrfy    cmp  #$20       ;  test if verify
     F693   F0 03         19275           beq  vrf10
                          19276   ;
     F695   4C F6CA       19277           jmp  sectsk
                          19278   ;
     F698                 19279   vrf10
                          19280   ;
     F698   20 F5E9       19281           jsr  chkblk     ;  get block checksum
     F69B   85 3A         19282           sta  chksum
                          19283   ;
     F69D   20 F78F       19284           jsr  bingcr     ;  convert to verify image
                          19285   ;
     F6A0   20 F50A       19286           jsr  dstrt
                          19287   ;
     F6A3   A0 BB         19288           ldy  #256-topwrt
     F6A5   B9 0100       19289   vrf15   lda  ovrbuf,y  ;  get char
     F6A8   50 FE         19290           bvc  *
     F6AA   B8            19291           clv
                          19292   ;
     F6AB   4D 1C01       19293           eor  data2      ;  test if same
     F6AE   D0 15         19294           bne  vrf20      ;  verify error
                          19295   ;
     F6B0   C8            19296           iny
     F6B1   D0 F2         19297           bne  vrf15      ;  next byte
                          19298   ;
                          19299   ;
     F6B3   B1 30         19300   vrf30   lda  (bufpnt),y ;  now do buffer
                          19301   ;
     F6B5   50 FE         19302           bvc  *
     F6B7   B8            19303           clv     ;  wait for char
                          19304   ;
     F6B8   4D 1C01       19305           eor  data2      ;  test if same
     F6BB   D0 08         19306           bne  vrf20      ;  error
                          19307   ;
     F6BD   C8            19308           iny
     F6BE   C0 FD         19309           cpy  #$fd       ;  dont test off bytes
     F6C0   D0 F1         19310           bne  vrf30
                          19311
                          19312
```

```
Error Addr  Code           Seq    Source statement

     F6C2   4C F418      19313            jmp  done        ;  verify ok
                         19314    ;
     F6C5   A9 07        19315    vrf20   lda  #7          ;  verify error
     F6C7   4C F969      19316            jmp  errr
                         19317    ;
                         19318    ;
     F6CA   20 F510      19319    sectsk  jsr  srch        ;  sector seek
     F6CD   4C F418      19320            jmp  done
                         19321    ;
                         19322    ;
                         19323    ;.end
                         19324            .include  lccbingcr
```

```
Error Addr  Code           Seq    Source statement

                          19326            .subttl 'lccbingcr.src'
                          19327    ;
                          19328    ;   fast binary to gcr
                          19329    ;
                          19330    ;
     F6D0   A9 00         19331    put4bg  lda  #0          ;  clear table
     F6D2   85 57         19332            sta  gtab+1
     F6D4   85 5A         19333            sta  gtab+4
                          19334    ;
     F6D6   A4 34         19335            ldy  gcrpnt
                          19336    ;
     F6D8   A5 52         19337            lda  btab
     F6DA   29 F0         19338            and  #$f0
     F6DC   4A            19339            lsr  a
     F6DD   4A            19340            lsr  a
     F6DE   4A            19341            lsr  a
     F6DF   4A            19342            lsr  a
     F6E0   AA            19343            tax
     F6E1   BD F77F       19344            lda  bgtab,x
                          19345    ;
     F6E4   0A            19346            asl  a
     F6E5   0A            19347            asl  a
     F6E6   0A            19348            asl  a
     F6E7   85 56         19349            sta  gtab
                          19350    ;
     F6E9   A5 52         19351            lda  btab
     F6EB   29 0F         19352            and  #$0f
     F6ED   AA            19353            tax
     F6EE   BD F77F       19354            lda  bgtab,x
                          19355    ;
     F6F1   6A            19356            ror  a
     F6F2   66 57         19357            ror  gtab+1
     F6F4   6A            19358            ror  a
     F6F5   66 57         19359            ror  gtab+1
                          19360    ;
     F6F7   29 07         19361            and  #$07
     F6F9   05 56         19362            ora  gtab
     F6FB   91 30         19363            sta  (bufpnt),y
                          19364    ;
     F6FD   C8            19365            iny
                          19366    ;
     F6FE   A5 53         19367            lda  btab+1
     F700   29 F0         19368            and  #$f0
     F702   4A            19369            lsr  a
     F703   4A            19370            lsr  a
     F704   4A            19371            lsr  a
     F705   4A            19372            lsr  a
     F706   AA            19373            tax
     F707   BD F77F       19374            lda  bgtab,x
                          19375    ;
```

```
Error Addr  Code           Seq    Source statement

     F70A   0A           19376            asl  a
     F70B   05 57        19377            ora  gtab+1
     F70D   85 57        19378            sta  gtab+1
                         19379  ;
                         19380  ;
     F70F   A5 53        19381            lda  btab+1
     F711   29 0F        19382            and  #$0f
     F713   AA           19383            tax
     F714   BD F77F      19384            lda  bgtab,x
                         19385  ;
     F717   2A           19386            rol  a
     F718   2A           19387            rol  a
     F719   2A           19388            rol  a
     F71A   2A           19389            rol  a
     F71B   85 58        19390            sta  gtab+2
                         19391  ;
     F71D   2A           19392            rol  a
     F71E   29 01        19393            and  #1
     F720   05 57        19394            ora  gtab+1
     F722   91 30        19395            sta  (bufpnt),y
                         19396  ;
     F724   C8           19397            iny
                         19398  ;
     F725   A5 54        19399            lda  btab+2
     F727   29 F0        19400            and  #$f0
     F729   4A           19401            lsr  a
     F72A   4A           19402            lsr  a
     F72B   4A           19403            lsr  a
     F72C   4A           19404            lsr  a
     F72D   AA           19405            tax
     F72E   BD F77F      19406            lda  bgtab,x
                         19407  ;
     F731   18           19408            clc
     F732   6A           19409            ror  a
     F733   05 58        19410            ora  gtab+2
     F735   91 30        19411            sta  (bufpnt),y
     F737   C8           19412            iny
                         19413  ;
     F738   6A           19414            ror  a
     F739   29 80        19415            and  #$80
     F73B   85 59        19416            sta  gtab+3
                         19417  ;
     F73D   A5 54        19418            lda  btab+2
     F73F   29 0F        19419            and  #$0f
     F741   AA           19420            tax
     F742   BD F77F      19421            lda  bgtab,x
     F745   0A           19422            asl  a
     F746   0A           19423            asl  a
     F747   29 7C        19424            and  #$7c
     F749   05 59        19425            ora  gtab+3
```

```
Error Addr  Code           Seq    Source statement

     F74B   85 59        19426            sta  gtab+3
                         19427    ;
     F74D   A5 55        19428            lda  btab+3
     F74F   29 F0        19429            and  #$f0
     F751   4A           19430            lsr  a
     F752   4A           19431            lsr  a
     F753   4A           19432            lsr  a
     F754   4A           19433            lsr  a
     F755   AA           19434            tax
     F756   BD F77F      19435            lda  bgtab,x
                         19436    ;
     F759   6A           19437            ror  a
     F75A   66 5A        19438            ror  gtab+4
     F75C   6A           19439            ror  a
     F75D   66 5A        19440            ror  gtab+4
     F75F   6A           19441            ror  a
     F760   66 5A        19442            ror  gtab+4
                         19443    ;
     F762   29 03        19444            and  #$03
     F764   05 59        19445            ora  gtab+3
     F766   91 30        19446            sta  (bufpnt),y
     F768   C8           19447            iny
     F769   D0 04        19448            bne  bing35
                         19449    ;
     F76B   A5 2F        19450            lda  savpnt+1
     F76D   85 31        19451            sta  bufpnt+1
                         19452    ;
                         19453    ;
     F76F   A5 55        19454    bing35  lda  btab+3
     F771   29 0F        19455            and  #$0f
     F773   AA           19456            tax
     F774   BD F77F      19457            lda  bgtab,x
     F777   05 5A        19458            ora  gtab+4
     F779   91 30        19459            sta  (bufpnt),y
     F77B   C8           19460            iny
     F77C   84 34        19461            sty  gcrpnt
     F77E   60           19462            rts
                         19463    ;
                         19464    ;
                         19465    ;
     F77F   0A           19466    bgtab   .byte    $0a
     F780   0B           19467            .byte    $0b
     F781   12           19468            .byte    $12
     F782   13           19469            .byte    $13
     F783   0E           19470            .byte    $0e
     F784   0F           19471            .byte    $0f
     F785   16           19472            .byte    $16
     F786   17           19473            .byte    $17
     F787   09           19474            .byte    $09
     F788   19           19475            .byte    $19
```

```
Error Addr  Code           Seq    Source statement

     F789   1A           19476            .byte    $1a
     F78A   1B           19477            .byte    $1b
     F78B   0D           19478            .byte    $0d
     F78C   1D           19479            .byte    $1d
     F78D   1E           19480            .byte    $1e
     F78E   15           19481            .byte    $15
                         19482    ;
                         19483    ;
                         19484    ;
                         19485    ;*******************************
                         19486    ;*
                         19487    ;*
                         19488    ;*       binary to gcr conversion
                         19489    ;*
                         19490    ;*
                         19491    ;*   does inplace conversion of
                         19492    ;*   buffer to gcr using overflow
                         19493    ;*   block
                         19494    ;*
                         19495    ;*
                         19496    ;*   creates write image
                         19497    ;*
                         19498    ;*     1 block id char
                         19499    ;*   256 data bytes
                         19500    ;*     1 check sum
                         19501    ;*     2 off bytes
                         19502    ;*   ---
                         19503    ;*   260 binary bytes
                         19504    ;*
                         19505    ;*  260 binary bytes >> 325 gcr
                         19506    ;*
                         19507    ;*  325 = 256 + 69 overflow
                         19508    ;*
                         19509    ;*
                         19510    ;*******************************
                         19511    ;*
     F78F   A9 00        19512    bingcr  lda  #0          ;  init pointers
     F791   85 30        19513            sta  bufpnt
     F793   85 2E        19514            sta  savpnt
     F795   85 36        19515            sta  bytcnt
                         19516    ;
     F797   A9 BB        19517            lda  #256-topwrt
     F799   85 34        19518            sta  gcrpnt      ;  start saving gcr here
                         19519    ;
     F79B   85 50        19520            sta  gcrflg      ;  buffer converted flag
                         19521    ;
     F79D   A5 31        19522            lda  bufpnt+1    ;  save buffer pointer
     F79F   85 2F        19523            sta  savpnt+1
                         19524    ;
     F7A1   A9 01        19525            lda  #>ovrbuf    ;  point at overflow
```

```
Error Addr  Code           Seq    Source statement

     F7A3   85 31        19526            sta  bufpnt+1
                         19527    ;
     F7A5   A5 47        19528            lda  dbid        ;  store data block id
     F7A7   85 52        19529            sta  btab        ;  and next 3 data bytes
                         19530    ;
     F7A9   A4 36        19531            ldy  bytcnt
                         19532    ;
     F7AB   B1 2E        19533            lda  (savpnt),y
     F7AD   85 53        19534            sta  btab+1
     F7AF   C8           19535            iny
                         19536    ;
     F7B0   B1 2E        19537            lda  (savpnt),y
     F7B2   85 54        19538            sta  btab+2
     F7B4   C8           19539            iny
                         19540    ;
     F7B5   B1 2E        19541            lda  (savpnt),y
     F7B7   85 55        19542            sta  btab+3
     F7B9   C8           19543            iny
                         19544    ;
     F7BA   84 36        19545    bing07  sty  bytcnt      ;  next byte to get
                         19546    ;
     F7BC   20 F6D0      19547            jsr  put4bg      ;  convert and store
                         19548    ;
     F7BF   A4 36        19549            ldy  bytcnt
                         19550    ;
     F7C1   B1 2E        19551            lda  (savpnt),y
     F7C3   85 52        19552            sta  btab
     F7C5   C8           19553            iny
     F7C6   F0 11        19554            beq  bing20
                         19555    ;
     F7C8   B1 2E        19556            lda  (savpnt),y
     F7CA   85 53        19557            sta  btab+1
     F7CC   C8           19558            iny
                         19559    ;
     F7CD   B1 2E        19560            lda  (savpnt),y
     F7CF   85 54        19561            sta  btab+2
     F7D1   C8           19562            iny
                         19563    ;
     F7D2   B1 2E        19564            lda  (savpnt),y
     F7D4   85 55        19565            sta  btab+3
     F7D6   C8           19566            iny
                         19567    ;
     F7D7   D0 E1        19568            bne  bing07      ;  jmp
                         19569    ;
                         19570    ;
     F7D9   A5 3A        19571    bing20  lda  chksum      ;  store chksum
     F7DB   85 53        19572            sta  btab+1
                         19573    ;
     F7DD   A9 00        19574            lda  #0          ;  store 0 off byte
     F7DF   85 54        19575            sta  btab+2
```

```
Error Addr  Code           Seq    Source statement

     F7E1   85 55        19576            sta  btab+3
                         19577    ;
     F7E3   4C F6D0      19578            jmp  put4bg     ;  convert and store and return
                         19579    ;
                         19580    ;
                         19581    ;
                         19582    ;.end
                         19583            .include  lccgcrbin
```

```
Error Addr  Code           Seq    Source statement

                          19585           .subttl 'lccgcrbin.src'
                          19586
            =00F8         19587   mask1=$f8
            =0007         19588   mask2=$07
            =00C0         19589   mask2x=$c0
            =003E         19590   mask3=$3e
            =0001         19591   mask4=$01
            =00F0         19592   mask4x=$f0
            =000F         19593   mask5=$0f
            =0080         19594   mask5x=$80
            =007C         19595   mask6=$7c
            =0003         19596   mask7=$03
            =00E0         19597   mask7x=$e0
            =001F         19598   mask8=$1f
                          19599   ;
                          19600   ;
                          19601   ;
                          19602   ;
                          19603   ; fast gcr to binary conversion
                          19604   ;
     F7E6   A4 34         19605   get4gb  ldy  gcrpnt
                          19606   ;
     F7E8   B1 30         19607           lda  (bufpnt),y
     F7EA   29 F8         19608           and  #mask1
     F7EC   4A            19609           lsr  a
     F7ED   4A            19610           lsr  a
     F7EE   4A            19611           lsr  a
     F7EF   85 56         19612           sta  gtab       ;  hi nibble
                          19613
     F7F1   B1 30         19614           lda  (bufpnt),y
     F7F3   29 07         19615           and  #mask2
     F7F5   0A            19616           asl  a
     F7F6   0A            19617           asl  a
     F7F7   85 57         19618           sta  gtab+1
                          19619
     F7F9   C8            19620           iny             ;  next byte
     F7FA   D0 06         19621           bne  xx05       ;  test for next buffer
     F7FC   A5 4E         19622           lda  nxtbf
     F7FE   85 31         19623           sta  bufpnt+1
     F800   A4 4F         19624           ldy  nxtpnt
                          19625
     F802   B1 30         19626   xx05    lda  (bufpnt),y
     F804   29 C0         19627           and  #mask2x
     F806   2A            19628           rol  a
     F807   2A            19629           rol  a
     F808   2A            19630           rol  a
     F809   05 57         19631           ora  gtab+1
     F80B   85 57         19632           sta  gtab+1
                          19633
     F80D   B1 30         19634           lda  (bufpnt),y
```

```
Error Addr  Code           Seq    Source statement

     F80F   29 3E        19635            and  #mask3
     F811   4A           19636            lsr  a
     F812   85 58        19637            sta  gtab+2
                         19638
     F814   B1 30        19639            lda  (bufpnt),y
     F816   29 01        19640            and  #mask4
     F818   0A           19641            asl  a
     F819   0A           19642            asl  a
     F81A   0A           19643            asl  a
     F81B   0A           19644            asl  a
     F81C   85 59        19645            sta  gtab+3
                         19646
     F81E   C8           19647            iny                         ;  next
                         19648
     F81F   B1 30        19649            lda  (bufpnt),y
     F821   29 F0        19650            and  #mask4x
     F823   4A           19651            lsr  a
     F824   4A           19652            lsr  a
     F825   4A           19653            lsr  a
     F826   4A           19654            lsr  a
     F827   05 59        19655            ora  gtab+3
     F829   85 59        19656            sta  gtab+3
                         19657
     F82B   B1 30        19658            lda  (bufpnt),y
     F82D   29 0F        19659            and  #mask5
     F82F   0A           19660            asl  a
     F830   85 5A        19661            sta  gtab+4
                         19662
     F832   C8           19663            iny                         ;  next byte
                         19664
     F833   B1 30        19665            lda  (bufpnt),y
     F835   29 80        19666            and  #mask5x
     F837   18           19667            clc
     F838   2A           19668            rol  a
     F839   2A           19669            rol  a
     F83A   29 01        19670            and  #1
     F83C   05 5A        19671            ora  gtab+4
     F83E   85 5A        19672            sta  gtab+4
                         19673
     F840   B1 30        19674            lda  (bufpnt),y
     F842   29 7C        19675            and  #mask6
     F844   4A           19676            lsr  a
     F845   4A           19677            lsr  a
     F846   85 5B        19678            sta  gtab+5
                         19679
     F848   B1 30        19680            lda  (bufpnt),y
     F84A   29 03        19681            and  #mask7
     F84C   0A           19682            asl  a
     F84D   0A           19683            asl  a
     F84E   0A           19684            asl  a
```

```
Error Addr  Code          Seq    Source statement

     F84F   85 5C       19685            sta  gtab+6
                        19686
     F851   C8          19687            iny
                        19688    ; test for overflow during write to binary conversion
     F852   D0 06       19689            bne  xx06
     F854   A5 4E       19690            lda  nxtbf
     F856   85 31       19691            sta  bufpnt+1
     F858   A4 4F       19692            ldy  nxtpnt
                        19693
     F85A   B1 30       19694    xx06    lda  (bufpnt),y
     F85C   29 E0       19695            and  #mask7x
     F85E   2A          19696            rol  a
     F85F   2A          19697            rol  a
     F860   2A          19698            rol  a
     F861   2A          19699            rol  a
     F862   05 5C       19700            ora  gtab+6
     F864   85 5C       19701            sta  gtab+6
                        19702
     F866   B1 30       19703            lda  (bufpnt),y
     F868   29 1F       19704            and  #mask8
     F86A   85 5D       19705            sta  gtab+7
     F86C   C8          19706            iny
                        19707
     F86D   84 34       19708            sty  gcrpnt
                        19709
                        19710
     F86F   A6 56       19711            ldx  gtab
     F871   BD F8A0     19712            lda  gcrhi,x
     F874   A6 57       19713            ldx  gtab+1
     F876   1D F8C0     19714            ora  gcrlo,x
     F879   85 52       19715            sta  btab
                        19716
     F87B   A6 58       19717            ldx  gtab+2
     F87D   BD F8A0     19718            lda  gcrhi,x
     F880   A6 59       19719            ldx  gtab+3
     F882   1D F8C0     19720            ora  gcrlo,x
     F885   85 53       19721            sta  btab+1
                        19722
     F887   A6 5A       19723            ldx  gtab+4
     F889   BD F8A0     19724            lda  gcrhi,x
     F88C   A6 5B       19725            ldx  gtab+5
     F88E   1D F8C0     19726            ora  gcrlo,x
     F891   85 54       19727            sta  btab+2
                        19728
     F893   A6 5C       19729            ldx  gtab+6
     F895   BD F8A0     19730            lda  gcrhi,x
     F898   A6 5D       19731            ldx  gtab+7
     F89A   1D F8C0     19732            ora  gcrlo,x
     F89D   85 55       19733            sta  btab+3
                        19734
```

```
Error Addr  Code           Seq    Source statement

     F89F   60           19735            rts
                         19736
     F8A0   FF           19737    gcrhi   .byte   $ff     ; error
     F8A1   FF           19738            .byte   $ff     ; error
     F8A2   FF           19739            .byte   $ff     ; error
     F8A3   FF           19740            .byte   $ff     ; error
     F8A4   FF           19741            .byte   $ff     ; error
     F8A5   FF           19742            .byte   $ff     ; error
     F8A6   FF           19743            .byte   $ff     ; error
     F8A7   FF           19744            .byte   $ff     ; error
     F8A8   FF           19745            .byte   $ff     ; error
     F8A9   80           19746            .byte   $80
     F8AA   00           19747            .byte   $00
     F8AB   10           19748            .byte   $10
     F8AC   FF           19749            .byte   $ff     ; error
     F8AD   C0           19750            .byte   $c0
     F8AE   40           19751            .byte   $40
     F8AF   50           19752            .byte   $50
     F8B0   FF           19753            .byte   $ff     ; error
     F8B1   FF           19754            .byte   $ff     ; error
     F8B2   20           19755            .byte   $20
     F8B3   30           19756            .byte   $30
     F8B4   FF           19757            .byte   $ff     ; error
     F8B5   F0           19758            .byte   $f0
     F8B6   60           19759            .byte   $60
     F8B7   70           19760            .byte   $70
     F8B8   FF           19761            .byte   $ff     ; error
     F8B9   90           19762            .byte   $90
     F8BA   A0           19763            .byte   $a0
     F8BB   B0           19764            .byte   $b0
     F8BC   FF           19765            .byte   $ff     ; error
     F8BD   D0           19766            .byte   $d0
     F8BE   E0           19767            .byte   $e0
     F8BF   FF           19768            .byte   $ff     ; error
                         19769    ;
                         19770    ;
                         19771    ;
     F8C0   FF           19772    gcrlo   .byte   $ff     ; error
     F8C1   FF           19773            .byte   $ff     ; error
     F8C2   FF           19774            .byte   $ff     ; error
     F8C3   FF           19775            .byte   $ff     ; error
     F8C4   FF           19776            .byte   $ff     ; error
     F8C5   FF           19777            .byte   $ff     ; error
     F8C6   FF           19778            .byte   $ff     ; error
     F8C7   FF           19779            .byte   $ff     ; error
     F8C8   FF           19780            .byte   $ff     ; error
     F8C9   08           19781            .byte   8
     F8CA   00           19782            .byte   $00
     F8CB   01           19783            .byte   1
     F8CC   FF           19784            .byte   $ff     ; error
```

```
Error Addr  Code           Seq    Source statement

     F8CD   0C           19785            .byte   $c
     F8CE   04           19786            .byte   4
     F8CF   05           19787            .byte   5
     F8D0   FF           19788            .byte   $ff     ; error
     F8D1   FF           19789            .byte   $ff     ; error
     F8D2   02           19790            .byte   2
     F8D3   03           19791            .byte   3
     F8D4   FF           19792            .byte   $ff     ; error
     F8D5   0F           19793            .byte   $f
     F8D6   06           19794            .byte   6
     F8D7   07           19795            .byte   7
     F8D8   FF           19796            .byte   $ff     ; error
     F8D9   09           19797            .byte   9
     F8DA   0A           19798            .byte   $a
     F8DB   0B           19799            .byte   $b
     F8DC   FF           19800            .byte   $ff     ; error
     F8DD   0D           19801            .byte   $d
     F8DE   0E           19802            .byte   $e
     F8DF   FF           19803            .byte   $ff     ; error
                         19804    ;
                         19805    ;
     F8E0   A9 00        19806    gcrbin  lda  #0          ;  setup pointers
     F8E2   85 34        19807            sta  gcrpnt
     F8E4   85 2E        19808            sta  savpnt
     F8E6   85 36        19809            sta  bytcnt
                         19810    ;
     F8E8   A9 01        19811            lda  #>ovrbuf
     F8EA   85 4E        19812            sta  nxtbf
                         19813    ;
     F8EC   A9 BA        19814            lda  #255-toprd
     F8EE   85 4F        19815            sta  nxtpnt
                         19816    ;
     F8F0   A5 31        19817            lda  bufpnt+1
     F8F2   85 2F        19818            sta  savpnt+1
                         19819    ;
     F8F4   20 F7E6      19820            jsr  get4gb
                         19821    ;
     F8F7   A5 52        19822            lda  btab
     F8F9   85 38        19823            sta  bid         ;  get header id
                         19824    ;
     F8FB   A4 36        19825            ldy  bytcnt
     F8FD   A5 53        19826            lda  btab+1
     F8FF   91 2E        19827            sta  (savpnt),y
     F901   C8           19828            iny
                         19829    ;
     F902   A5 54        19830            lda  btab+2
     F904   91 2E        19831            sta  (savpnt),y
     F906   C8           19832            iny
                         19833    ;
     F907   A5 55        19834            lda  btab+3
```

```
Error Addr  Code           Seq    Source statement

     F909   91 2E        19835            sta  (savpnt),y
     F90B   C8           19836            iny
                         19837    ;
     F90C   84 36        19838    gcrb10  sty  bytcnt
                         19839    ;
     F90E   20 F7E6      19840            jsr  get4gb
                         19841    ;
     F911   A4 36        19842            ldy  bytcnt
                         19843    ;
     F913   A5 52        19844            lda  btab
     F915   91 2E        19845            sta  (savpnt),y
     F917   C8           19846            iny
     F918   F0 11        19847            beq  gcrb20      ;  test if done yet
                         19848    ;
     F91A   A5 53        19849            lda  btab+1
     F91C   91 2E        19850            sta  (savpnt),y
     F91E   C8           19851            iny
                         19852
     F91F   A5 54        19853            lda  btab+2
     F921   91 2E        19854            sta  (savpnt),y
     F923   C8           19855            iny
                         19856    ;
     F924   A5 55        19857            lda  btab+3
     F926   91 2E        19858            sta  (savpnt),y
     F928   C8           19859            iny
                         19860    ;
     F929   D0 E1        19861            bne  gcrb10      ;  jmp
                         19862    ;
     F92B                19863    gcrb20
     F92B   A5 53        19864            lda  btab+1
     F92D   85 3A        19865            sta  chksum
     F92F   A5 2F        19866            lda  savpnt+1    ; restore buffer pointer
     F931   85 31        19867            sta  bufpnt+1
                         19868    ;
     F933   60           19869            rts
                         19870    ;.end
                         19871            .include  lccconhdr
```

```
Error Addr  Code           Seq    Source statement

                          19873            .subttl 'lccconhdr.src'
                          19874   ;
                          19875   ;
                          19876   ;
                          19877   ;    *conhdr
                          19878   ;
                          19879   ;    convert header
                          19880   ;    into gcr search image
                          19881   ;    and place in stab
                          19882   ;
                          19883   ;   image contains :
                          19884   ;
                          19885   ;   00 id id tr sc cs hbid
                          19886   ;
                          19887   ;
     F934   A5 31         19888   conhdr  lda  bufpnt+1   ; save buffer pointer
     F936   85 2F         19889           sta  savpnt+1
                          19890   ;
     F938   A9 00         19891           lda  #>stab
     F93A   85 31         19892           sta  bufpnt+1
                          19893   ;
     F93C   A9 24         19894           lda  #<stab
     F93E   85 34         19895           sta  gcrpnt
                          19896   ;
     F940   A5 39         19897           lda  hbid
     F942   85 52         19898           sta  btab
                          19899   ;
     F944   A5 1A         19900           lda  header+4
     F946   85 53         19901           sta  btab+1
                          19902   ;
     F948   A5 19         19903           lda  header+3
     F94A   85 54         19904           sta  btab+2
                          19905   ;
     F94C   A5 18         19906           lda  header+2
     F94E   85 55         19907           sta  btab+3
                          19908   ;
     F950   20 F6D0       19909           jsr  put4bg
                          19910   ;
     F953   A5 17         19911           lda  header+1
     F955   85 52         19912           sta  btab
                          19913   ;
     F957   A5 16         19914           lda  header
     F959   85 53         19915           sta  btab+1
                          19916   ;
     F95B   A9 00         19917           lda  #0
     F95D   85 54         19918           sta  btab+2
     F95F   85 55         19919           sta  btab+3
                          19920   ;
     F961   20 F6D0       19921           jsr  put4bg
                          19922
```

```
Error Addr  Code           Seq    Source statement

     F964   A5 2F        19923            lda  savpnt+1   ; restore buffer pointer
     F966   85 31        19924            sta  bufpnt+1
                         19925    ;
     F968   60           19926            rts
                         19927    ;
                         19928    ;
                         19929    ;.end
                         19930            .include  lccutil
```

```
Error Addr  Code           Seq    Source statement

                          19932            .subttl 'lccutil.src'
                          19933   ;
                          19934   ;
                          19935   ;  * utility routines
                          19936   ;
                          19937   ;
     F969   A4 3F         19938   errr     ldy  jobn       ;  return  job code
     F96B   99 0000       19939            sta  jobs,y
                          19940   ;
     F96E   A5 50         19941            lda  gcrflg     ;  test if buffer left gcr
     F970   F0 03         19942            beq  errr10     ;  no
                          19943   ;
     F972   20 F5F2       19944            jsr  wtobin     ;  convert back to binary
                          19945   ;
     F975                 19946   errr10
     F975   20 F98F       19947            jsr  trnoff     ;  start timeout on drive
                          19948   ;
     F978   A6 49         19949            ldx  savsp
     F97A   9A            19950            txs             ;  reset stack pointer
                          19951   ;
     F97B   4C F2BE       19952            jmp  top        ;  back to the top
                          19953   ;
                          19954   ;
                          19955   ;
     F97E   A9 A0         19956   turnon   lda  #$a0       ;  turn on drive
                          19957   ; drvst=acel and on
     F980   85 20         19958            sta  drvst
                          19959   ;
                          19960   ;
     F982   AD 1C00       19961            lda  dskcnt     ;  turn motor on and select drive
     F985   09 04         19962            ora  #$04       ;  turn motor on
     F987   8D 1C00       19963            sta  dskcnt
                          19964   ;
     F98A   A9 32         19965            lda  #50        ;  delay  .4 sec *** rom ds 04/17/85 ***
     F98C   85 48         19966            sta  acltim
                          19967   ;
     F98E   60            19968            rts
                          19969   ;
                          19970   ;
                          19971   ;
     F98F   A6 3E         19972   trnoff   ldx  cdrive     ;  start time out of current drive
     F991   A5 20         19973            lda  drvst      ; status=timeout
     F993   09 10         19974            ora  #$10
     F995   85 20         19975            sta  drvst
                          19976   ;
     F997   4C A62B       19977            jmp  ptch20     ; setup timers for timeout *rom-05ds 01/22/85*
     F99A   EA            19978            nop             ; fill
     F99B   EA            19979            nop             ; fill
                          19980
                          19981   ;        lda  #255       ;  255*.025s time out
```

```
Error Addr  Code          Seq   Source statement

                        19982   ;       sta  acltim
                        19983   ;
                        19984   ;       rts
                        19985   ;
                        19986   ;
                        19987   ;
                        19988   ;.end
                        19989           .include  lccend
```

```
Error Addr  Code           Seq    Source statement

                          19991            .subttl 'lccend.src'
                          19992    ;
                          19993    ;
                          19994    ;
                          19995    ;   motor and stepper control
                          19996    ;
                          19997    ;
                          19998    ;   irq into controller every  8.224 ms
     F99C                 19999    end
     F99C   AD 1C07       20000            lda  t1hl2       ; set irq timer
     F99F   8D 1C05       20001            sta  t1hc2
                          20002    ;
     F9A2   AD 1C00       20003            lda  dskcnt
     F9A5   29 10         20004            and  #$10        ;  test write proctect
     F9A7   C5 1E         20005            cmp  lwpt
     F9A9   85 1E         20006            sta  lwpt        ;  change ?
                          20007
                          20008    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          20009
     F9AB   4C A634       20010            jmp  ptch21      ;  do auto on code *** rom ds 01/22/85 ***
     F9AE   EA            20011            nop              ;  fill
     F9AF   EA            20012            nop              ;  fill
     F9B0   EA            20013            nop              ;  fill
                          20014
                          20015    ;       beq  end002      ;  no
                          20016
                          20017    ;       lda  #1          ;  yes, set flag
                          20018    ;       sta  wpsw
                          20019    ;
                          20020
     F9B1                 20021    rtch21                   ; return
                          20022
                          20023    ;<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
                          20024
     F9B1   AD 02FE       20025    end002  lda  phase       ;  test for phase offset
     F9B4   F0 15         20026            beq  end40       ;
                          20027    ;
     F9B6   C9 02         20028            cmp  #2
     F9B8   D0 07         20029            bne  end003
                          20030    ;
     F9BA   A9 00         20031            lda  #0          ; phase <-- 0
     F9BC   8D 02FE       20032            sta  phase
     F9BF   F0 0A         20033            beq  end40
                          20034    ;
     F9C1   85 4A         20035    end003  sta  steps
     F9C3   A9 02         20036            lda  #2          ; phase <-- 2
     F9C5   8D 02FE       20037            sta  phase
     F9C8   4C FA2E       20038            jmp  dostep
                          20039    ;
     F9CB   A6 3E         20040    end40   ldx  cdrive      ;  work on active drive only
```

```
Error Addr  Code           Seq    Source statement

     F9CD   30 07        20041            bmi  end33x     ;  no active drive
                         20042    ;
     F9CF   A5 20        20043            lda  drvst      ;  test if motor on
     F9D1   A8           20044            tay
     F9D2   C9 20        20045            cmp  #$20       ;  test if anything to do
     F9D4   D0 03        20046            bne  end10      ;  something here
                         20047    ;
     F9D6   4C FABE      20048    end33x  jmp  end33      ;  motor just running
                         20049    ;
     F9D9   C6 48        20050    end10   dec  acltim     ;  dec timer
     F9DB   D0 1D        20051            bne  end30
                         20052    ;
     F9DD   98           20053            tya             ;  test if acel
     F9DE   10 04        20054            bpl  end20
                         20055    ;
                         20056    ;
     F9E0   29 7F        20057            and  #$7f       ;  over, clear acel bit
     F9E2   85 20        20058            sta  drvst
                         20059    ;
     F9E4   29 10        20060    end20   and  #$10       ;  test if time out state
     F9E6   F0 12        20061            beq  end30
                         20062
     F9E8   C6 35        20063            dec  acltim2    ;  timer2 *rom-05ds 01/22/85*
     F9EA   D0 0E        20064            bne  end30
     F9EC   EA           20065            nop             ;  fill
     F9ED   20 8770      20066            jsr  motoff     ;  off the motor, ok...
                         20067    ;
                         20068    ;       lda  dskcnt     ;
                         20069    ;       and  #$ff-$04   ;  turnoff motor
                         20070    ;       sta  dskcnt
                         20071    ;
                         20072    ;
     F9F0   A9 FF        20073            lda  #$ff       ;  no active drive now
     F9F2   85 3E        20074            sta  cdrive
                         20075    ;
     F9F4   A9 00        20076            lda  #0         ;  drive inactive
     F9F6   85 20        20077            sta  drvst      ;  clear on bit and timout
     F9F8   F0 DC        20078            beq  end33x
                         20079    ;
     F9FA   98           20080    end30   tya             ;  test if step needed
     F9FB   29 40        20081            and  #$40
     F9FD   D0 03        20082            bne  end30x     ;  stepping
                         20083    ;
     F9FF   4C FABE      20084            jmp  end33
                         20085    ;
                         20086    ;
     FA02   6C 0062      20087    end30x  jmp  (nxtst)    ; goto proper stepper state
                         20088    ;
     FA05   A5 4A        20089    inact   lda  steps      ;  get abs(steps)
     FA07   10 05        20090            bpl  inac10
```

```
Error Addr  Code           Seq    Source statement

                          20091   ;
      FA09   49 FF        20092           eor  #$ff
      FA0B   18           20093           clc
      FA0C   69 01        20094           adc  #1
                          20095   ;
      FA0E   C5 64        20096   inac10  cmp  minstp     ;  test if we can accel
      FA10   B0 0A        20097           bcs  inac20     ;  too small
                          20098   ;
      FA12   A9 3B        20099           lda  #<short    ; short step mode
      FA14   85 62        20100           sta  nxtst
      FA16   A9 FA        20101           lda  #>short
      FA18   85 63        20102           sta  nxtst+1
      FA1A   D0 12        20103           bne  dostep
                          20104   ;
      FA1C                20105   inac20                  ;  calc the # of run steps
      FA1C   E5 5E        20106           sbc  as
      FA1E   E5 5E        20107           sbc  as
      FA20   85 61        20108           sta  rsteps
                          20109   ;
      FA22   A5 5E        20110           lda  as
      FA24   85 60        20111           sta  aclstp     ;  set  # of accel steps
      FA26   A9 7B        20112           lda  #<ssacl
      FA28   85 62        20113           sta  nxtst
      FA2A   A9 FA        20114           lda  #>ssacl
      FA2C   85 63        20115           sta  nxtst+1
                          20116   ;
      FA2E   A5 4A        20117   dostep  lda  steps
      FA30   10 31        20118           bpl  stpin
                          20119   ;
                          20120   ;---------rom05-bc 09/12/84--------
                          20121   ;stpout inc steps
                          20122   ;       ldx dskcnt
                          20123   ;       dex
                          20124   ;       jmp stp
                          20125
      FA32   4C FF45      20126   stpout  jmp  patch9     ; check track 0 (photo sensor)
      FA35   EA           20127           nop             ;
      FA36   EA           20128           nop             ;
      FA37   EA           20129           nop             ;
      FA38   4C FA69      20130   pppppp  jmp  stp        ; goto step
                          20131   ;----------------------------
                          20132   ;
      FA3B   A5 4A        20133   short   lda  steps      ; step end ?
      FA3D   D0 EF        20134           bne  dostep     ; no
                          20135   ;
      FA3F   A9 4E        20136           lda  #<setle    ; settle
      FA41   85 62        20137           sta  nxtst
      FA43   A9 FA        20138           lda  #>setle
      FA45   85 63        20139           sta  nxtst+1
      FA47   A9 05        20140           lda  #5         ;  settle time (5*8=40ms)
```

```
Error Addr  Code           Seq    Source statement

     FA49   85 60        20141            sta  aclstp
     FA4B   4C FABE      20142            jmp  end33
                         20143  ;
     FA4E   C6 60        20144  setle     dec  aclstp      ;  settle end ?
     FA50   D0 6C        20145            bne  end33       ;  no
                         20146
     FA52   A5 20        20147            lda  drvst
     FA54   29 BF        20148            and  #$ff-$40
     FA56   85 20        20149            sta  drvst
                         20150
     FA58   A9 05        20151            lda  #<inact
     FA5A   85 62        20152            sta  nxtst
     FA5C   A9 FA        20153            lda  #>inact
     FA5E   85 63        20154            sta  nxtst+1
     FA60   4C FABE      20155            jmp  end33
                         20156  ;
     FA63   C6 4A        20157  stpin     dec  steps
     FA65   AE 1C00      20158            ldx  dskcnt
     FA68   E8           20159            inx
                         20160  ;
     FA69   8A           20161  stp       txa
     FA6A   29 03        20162            and  #3
     FA6C   85 4B        20163            sta  tmp
     FA6E   AD 1C00      20164            lda  dskcnt
     FA71   29 FC        20165            and  #$ff-$03    ;  mask out old
     FA73   05 4B        20166            ora  tmp
     FA75   8D 1C00      20167            sta  dskcnt
     FA78   4C FABE      20168            jmp  end33
                         20169  ;
     FA7B                20170  ssacl                     ;  sub acel factor
     FA7B   38           20171            sec
     FA7C   AD 1C07      20172            lda  t1hl2
     FA7F   E5 5F        20173            sbc  af
     FA81   8D 1C05      20174            sta  t1hc2
                         20175
     FA84   C6 60        20176            dec  aclstp
     FA86   D0 0C        20177            bne  ssa10
                         20178  ;
     FA88   A5 5E        20179            lda  as
     FA8A   85 60        20180            sta  aclstp
                         20181  ;
     FA8C   A9 97        20182            lda  #<ssrun
     FA8E   85 62        20183            sta  nxtst
     FA90   A9 FA        20184            lda  #>ssrun
     FA92   85 63        20185            sta  nxtst+1
                         20186  ;
     FA94   4C FA2E      20187  ssa10     jmp  dostep
                         20188  ;
     FA97   C6 61        20189  ssrun     dec  rsteps
     FA99   D0 F9        20190            bne  ssa10
```

```
Error Addr  Code           Seq    Source statement

                          20191   ;
      FA9B   A9 A5        20192           lda  #<ssdec
      FA9D   85 62        20193           sta  nxtst
      FA9F   A9 FA        20194           lda  #>ssdec
      FAA1   85 63        20195           sta  nxtst+1
      FAA3   DO EF        20196           bne  ssa10
                          20197   ;
      FAA5                20198   ssdec                   ;  decel
      FAA5   AD 1C07      20199           lda  t1hl2
      FAA8   18           20200           clc
      FAA9   65 5F        20201           adc  af
      FAAB   8D 1C05      20202           sta  t1hc2
                          20203   ;
      FAAE   C6 6O        20204           dec  aclstp
      FABO   DO E2        20205           bne  ssa10
                          20206   ;
      FAB2   A9 4E        20207           lda  #<setle    ;  goto settle mode
      FAB4   85 62        20208           sta  nxtst
      FAB6   A9 FA        20209           lda  #>setle
      FAB8   85 63        20210           sta  nxtst+1
                          20211
      FABA   A9 O5        20212           lda  #5
      FABC   85 6O        20213           sta  aclstp     ;  settle timer
                          20214   ;
                          20215   ;
      FABE   AD 1COC      20216   end33   lda  pcr2       ;  disable s.o. to 6502
      FAC1   29 FD        20217           and  #$ff-$02
      FAC3   8D 1COC      20218           sta  pcr2
                          20219   ;
      FAC6   6O           20220           rts
                          20221   ;.end
                          20222           .include  lccfmt1
```

```
Error Addr  Code           Seq    Source statement

                          20224           .subttl 'lccfmt1.src'
                          20225   ;
            =0620         20226   fmtvar   =  $620        ;  put format vars in jump buffer
            =0620         20227   cnt      =  fmtvar
            =0621         20228   num      =  fmtvar+1
            =0623         20229   trys     =  fmtvar+3
            =0624         20230   tral     =  fmtvar+4
            =0626         20231   dtrck    =  fmtvar+6
            =0627         20232   remdr    =  fmtvar+7
            =0628         20233   sect     =  fmtvar+8
                          20234   ;
                          20235   ;
                          20236   ;
                          20237   ;*  format routine for lcc
                          20238   ;*
                          20239   ;*
                          20240   ;*
                          20241   ;
                          20242   ;
                          20243   ;
     FAC7                 20244   code
     FAC7   A5 51         20245   formt    lda  ftnum      ;  test if formatting begun
     FAC9   10 2A         20246            bpl  1213       ;  yes
                          20247   ;
     FACB   A6 3D         20248            ldx  drive      ;  no,start up by bumping
     FACD   A9 60         20249            lda  #$60       ;  status=stepping
     FACF   95 20         20250            sta  drvst,x
                          20251   ;
     FAD1   A9 01         20252            lda  #1         ;  drive track =1
     FAD3   95 22         20253            sta  drvtrk,x
     FAD5   85 51         20254            sta  ftnum      ;  start on track 1
                          20255   ;
     FAD7   A9 A4         20256            lda  #256-92    ;  bump back 45 steps
     FAD9   85 4A         20257            sta  steps
                          20258   ;
     FADB   AD 1C00       20259            lda  dskcnt     ; set phase a
     FADE   29 FC         20260            and  #$ff-$03
     FAE0   8D 1C00       20261            sta  dskcnt
                          20262   ;
                          20263   ;
     FAE3   A9 0A         20264            lda  #10        ;  10 errors allowed
     FAE5   8D 0620       20265            sta  cnt
                          20266   ;
     FAE8   A9 A0         20267            lda  #<4000     ;  first guess at track size
     FAEA   8D 0621       20268            sta  num
     FAED   A9 0F         20269            lda  #>4000
     FAEF   8D 0622       20270            sta  num+1
                          20271   ;
     FAF2   4C F99C       20272            jmp  end        ;  back to controller
                          20273
```

```
Error Addr  Code           Seq    Source statement

                          20274   ;
                          20275   ;
     FAF5   A0 00         20276   l213    ldy  #0          ;  test if on right track number
     FAF7   D1 32         20277           cmp  (hdrpnt),y
     FAF9   F0 05         20278           beq  l214
                          20279   ;
     FAFB   91 32         20280           sta  (hdrpnt),y  ;  goto right track
     FAFD   4C F99C       20281           jmp  end
                          20282   ;
                          20283   ;
     FB00   AD 1C00       20284   l214    lda  dskcnt      ;  test for write protect
     FB03   29 10         20285           and  #$10
     FB05   D0 05         20286           bne  topp        ;  its ok
                          20287   ;
     FB07   A9 08         20288           lda  #8          ;  write protect error
     FB09   4C FDD3       20289           jmp  fmterr
                          20290   ;
     FB0C   20 FDA3       20291   topp    jsr  synclr      ;  erase track with sync
                          20292   ;
     FB0F   20 FDC3       20293           jsr  wrtnum      ;  write out num syncs
                          20294   ;
     FB12   A9 55         20295           lda  #$55        ;  write out num non sync
     FB14   8D 1C01       20296           sta  data2
                          20297   ;
     FB17   20 FDC3       20298           jsr  wrtnum
                          20299   ;
     FB1A   20 FE00       20300           jsr  kill        ;  kill write
                          20301   ;
     FB1D   20 F556       20302           jsr  sync        ;  find sync
                          20303
     FB20   A9 40         20304           lda  #$40        ;  set timer mode
     FB22   0D 180B       20305           ora  acr1
     FB25   8D 180B       20306           sta  acr1
                          20307   ;
     FB28   A9 62         20308           lda  #100-2      ;  set up 100us timer
     FB2A   8D 1806       20309           sta  t1ll1       ;  cont mode timer
     FB2D   A9 00         20310           lda  #0
     FB2F   8D 1807       20311           sta  t1hl1       ;  hi latch
     FB32   8D 1805       20312           sta  t1hc1       ;  get attention of '22
                          20313   ;
                          20314   ;
                          20315   ;
     FB35   A0 00         20316           ldy  #0          ;  time the sync and nonsync segments
     FB37   A2 00         20317           ldx  #0
                          20318   ;
     FB39   2C 1C00       20319   fwait   bit  dskcnt      ;  wait for sync
     FB3C   30 FB         20320           bmi  fwait
                          20321   ;
     FB3E   2C 1C00       20322   fwait2  bit  dskcnt      ;  wait for no sync
     FB41   10 FB         20323           bpl  fwait2
```

```
Error Addr  Code           Seq    Source statement

                          20324   ;
     FB43   AD 1804       20325   f000    lda  t1lc1       ;  reset ifr
                          20326   ;
     FB46   2C 1C00       20327   f001    bit  dskcnt      ;  time nonsync area
     FB49   10 11         20328           bpl  f005        ;  time until sync found
                          20329   ;
     FB4B   AD 180D       20330           lda  ifr1        ;  test for time out
     FB4E   0A            20331           asl  a
     FB4F   10 F5         20332           bpl  f001        ;  not yet
                          20333
     FB51   E8            20334           inx              ;  .x is lsb
     FB52   D0 EF         20335           bne  f000
     FB54   C8            20336           iny              ;  .y is msb
     FB55   D0 EC         20337           bne  f000
                          20338   ;
     FB57   A9 02         20339           lda  #tolong     ;  cant find sync
     FB59   4C FDD3       20340           jmp  fmterr
                          20341   ;
     FB5C   86 71         20342   f005    stx  t2          ;  save time
     FB5E   84 72         20343           sty  t2+1
                          20344   ;
     FB60   A2 00         20345           ldx  #0          ;  time sync area
     FB62   A0 00         20346           ldy  #0
                          20347   ;
     FB64   AD 1804       20348   f006    lda  t1lc1       ;  reset ifr
                          20349   ;
     FB67   2C 1C00       20350   f007    bit  dskcnt      ;  test for  no sync
     FB6A   30 11         20351           bmi  f009
                          20352   ;
     FB6C   AD 180D       20353           lda  ifr1        ;  test for time out
     FB6F   0A            20354           asl  a
     FB70   10 F5         20355           bpl  f007
                          20356   ;
     FB72   E8            20357           inx              ;  count up another 100us
     FB73   D0 EF         20358           bne  f006
     FB75   C8            20359           iny              ;  msb
     FB76   D0 EC         20360           bne  f006
                          20361
     FB78   A9 02         20362           lda  #tolong     ;  cant be this long
     FB7A   4C FDD3       20363           jmp  fmterr
                          20364   ;
                          20365   ;
                          20366   ;* now calc the difference between
                          20367   ;* sync and nonsync and adjust
                          20368   ;* num accordingly
                          20369   ;
     FB7D   38            20370   f009    sec              ;  t1-t2
     FB7E   8A            20371           txa
     FB7F   E5 71         20372           sbc  t2
     FB81   AA            20373           tax
```

```
Error Addr  Code          Seq    Source statement

     FB82   85 70        20374            sta  t1
                         20375  ;
     FB84   98           20376            tya
     FB85   E5 72        20377            sbc  t2+1
     FB87   A8           20378            tay
     FB88   85 71        20379            sta  t1+1
                         20380  ;
     FB8A   10 0B        20381            bpl  f013       ;  get abs(t1-t2)
                         20382  ;
     FB8C   49 FF        20383            eor  #$ff       ;  make +
     FB8E   A8           20384            tay
     FB8F   8A           20385            txa
     FB90   49 FF        20386            eor  #$ff
     FB92   AA           20387            tax
     FB93   E8           20388            inx
     FB94   D0 01        20389            bne  f013
     FB96   C8           20390            iny
                         20391  ;
     FB97   98           20392  f013      tya             ;  test if abs(t1-t2)<4, that is close enough
     FB98   D0 04        20393            bne  f014       ;  msb must be 0
                         20394  ;
     FB9A   E0 04        20395            cpx  #4         ;  test lsb < 4
     FB9C   90 18        20396            bcc  count      ;  its there
                         20397  ;
     FB9E   06 70        20398  f014      asl  t1         ;  num=num+(diff/2)
     FBA0   26 71        20399            rol  t1+1
                         20400  ;
     FBA2   18           20401            clc
     FBA3   A5 70        20402            lda  t1
     FBA5   6D 0621      20403            adc  num
     FBA8   8D 0621      20404            sta  num
                         20405  ;
     FBAB   A5 71        20406            lda  t1+1
     FBAD   6D 0622      20407            adc  num+1
     FBB0   8D 0622      20408            sta  num+1
                         20409  ;
     FBB3   4C FB0C      20410            jmp  topp       ;  try again sam
                         20411  ;
                         20412  ;
     FBB6   A2 00        20413  count     ldx  #0         ;  now count #bytes in data segment
     FBB8   A0 00        20414            ldy  #0
     FBBA   B8           20415            clv
                         20416  ;
     FBBB   AD 1C00      20417  cnt10     lda  dskcnt     ;  test for sync
     FBBE   10 0E        20418            bpl  cnt20      ;  found sync
     FBC0   50 F9        20419            bvc  cnt10      ;  test if byte time
                         20420  ;
     FBC2   B8           20421            clv             ;  yes, count it
     FBC3   E8           20422            inx
     FBC4   D0 F5        20423            bne  cnt10      ;  keep counting
```

```
Error Addr  Code           Seq    Source statement

     FBC6   C8           20424            iny
     FBC7   D0 F2        20425            bne  cnt10       ;  to many ?
                         20426   ;
     FBC9   A9 03        20427            lda  #tomany     ;  tomany counts
     FBCB   4C FDD3      20428            jmp  fmterr
                         20429   ;
     FBCE   8A           20430   cnt20    txa              ;  #bytes=count*2
     FBCF   0A           20431            asl  a
     FBD0   8D 0625      20432            sta  tra1+1
                         20433   ;
     FBD3   98           20434            tya
     FBD4   2A           20435            rol  a
     FBD5   8D 0624      20436            sta  tra1
                         20437   ;
     FBD8   A9 BF        20438            lda  #$ff-$40    ;  clear cont mode
     FBDA   2D 180B      20439            and  acr1
     FBDD   8D 180B      20440            sta  acr1
                         20441   ;
                         20442   ;
                         20443   ;.end
                         20444            .include  lccfmt2
```

```
Error Addr  Code           Seq    Source statement

                          20446            .subttl 'lccfmt2.src'
                          20447    ;
                          20448    ;
                          20449    ;
     FBE0   A9 66         20450    ds08    lda  #$66        ;  min block size 282*5/4 -256=85
                          20451    ;
     FBE2   8D 0626       20452            sta  dtrck
                          20453    ;
     FBE5   A6 43         20454            ldx  sectr       ;  total bytes= min*#secors
     FBE7   A0 00         20455            ldy  #0
     FBE9   98            20456            tya
                          20457    ;
     FBEA   18            20458    ds10    clc              ;  min*#sectors
     FBEB   6D 0626       20459            adc  dtrck
     FBEE   90 01         20460            bcc  ds14
                          20461    ;
     FBF0   C8            20462            iny
                          20463    ;
     FBF1   C8            20464    ds14    iny
     FBF2   CA            20465            dex
     FBF3   D0 F5         20466            bne  ds10
                          20467    ;
     FBF5   49 FF         20468            eor  #$ff        ;  get 2s comp
     FBF7   38            20469            sec
     FBF8   69 00         20470            adc  #0
                          20471    ;
     FBFA   18            20472            clc
     FBFB   6D 0625       20473            adc  tral+1
     FBFE   B0 03         20474            bcs  ds15
                          20475    ;
     FC00   CE 0624       20476            dec  tral
                          20477    ;
     FC03   AA            20478    ds15    tax
     FC04   98            20479            tya
     FC05   49 FF         20480            eor  #$ff
     FC07   38            20481            sec
     FC08   69 00         20482            adc  #0
     FC0A   18            20483            clc
     FC0B   6D 0624       20484            adc  tral
                          20485    ;
     FC0E   10 05         20486            bpl  ds17
                          20487    ;
     FC10   A9 04         20488            lda  #tobig      ;  not enough space
     FC12   4C FDD3       20489            jmp  fmterr
                          20490    ;
     FC15   A8            20491    ds17    tay
     FC16   8A            20492            txa
     FC17   A2 00         20493            ldx  #0
                          20494    ;
     FC19   38            20495    ds20    sec
```

```
Error Addr  Code          Seq   Source statement

     FC1A   E5 43       20496           sbc  sectr
     FC1C   B0 03       20497           bcs  ds22
                        20498   ;
     FC1E   88          20499           dey
     FC1F   30 03       20500           bmi  ds30
                        20501   ;
     FC21   E8          20502   ds22    inx
     FC22   D0 F5       20503           bne  ds20
                        20504   ;
     FC24   8E 0626     20505   ds30    stx  dtrck
                        20506
                        20507   ;-----rom05-bc---09/12/84------
     FC27   E0 04       20508           cpx  #gap2      ;  test for min size
                        20509   ;-------------------------------
                        20510
     FC29   B0 05       20511           bcs  ds32
                        20512   ;
     FC2B   A9 05       20513           lda  #tosmal    ;  gap2 to small
     FC2D   4C FDD3     20514           jmp  fmterr
                        20515   ;
     FC30   18          20516   ds32    clc
     FC31   65 43       20517           adc  sectr
     FC33   8D 0627     20518           sta  remdr      ;  get remaider size
                        20519   ;
                        20520   ;
                        20521   ;
                        20522   ;
                        20523   ;
                        20524   ;  create header images
                        20525   ;
                        20526   ;
     FC36   A9 00       20527           lda  #0
     FC38   8D 0628     20528           sta  sect
                        20529   ;
     FC3B   A0 00       20530           ldy  #0
     FC3D   A6 3D       20531           ldx  drive
                        20532   ;
     FC3F   A5 39       20533   mak10   lda  hbid      ;  hbid cs s t id id 0f 0f
     FC41   99 0300     20534           sta  buff0,y
     FC44   C8          20535           iny
                        20536   ;
     FC45   C8          20537           iny             ;  skip checksum
                        20538   ;
     FC46   AD 0628     20539           lda  sect       ;  store sector #
     FC49   99 0300     20540           sta  buff0,y
     FC4C   C8          20541           iny
                        20542   ;
     FC4D   A5 51       20543           lda  ftnum      ;  store track #
     FC4F   99 0300     20544           sta  buff0,y
     FC52   C8          20545           iny
```

```
Error Addr  Code           Seq    Source statement

                          20546   ;
     FC53   B5 13         20547           lda  dskid+1,x  ;  store id low
     FC55   99 0300       20548           sta  buff0,y
     FC58   C8            20549           iny
                          20550   ;
     FC59   B5 12         20551           lda  dskid,x    ;  store id hi
     FC5B   99 0300       20552           sta  buff0,y
     FC5E   C8            20553           iny
                          20554   ;
     FC5F   A9 0F         20555           lda  #$0f       ;  store gap1 bytes
     FC61   99 0300       20556           sta  buff0,y
     FC64   C8            20557           iny
     FC65   99 0300       20558           sta  buff0,y
     FC68   C8            20559           iny
                          20560   ;
     FC69   A9 00         20561           lda  #0         ; create checksum
     FC6B   59 02FA       20562           eor  buff0-6,y
     FC6E   59 02FB       20563           eor  buff0-5,y
     FC71   59 02FC       20564           eor  buff0-4,y
     FC74   59 02FD       20565           eor  buff0-3,y
                          20566   ;
     FC77   99 02F9       20567           sta  buff0-7,y  ;  store checksum
                          20568   ;
                          20569   ;
     FC7A   EE 0628       20570           inc  sect       ;  goto next sector
                          20571   ;
     FC7D   AD 0628       20572           lda  sect       ;  test if done yet
     FC80   C5 43         20573           cmp  sectr
     FC82   90 BB         20574           bcc  mak10      ;  more to do
                          20575   ;
     FC84   98            20576           tya             ;  save block size
     FC85   48            20577           pha
                          20578   ;
                          20579   ;
                          20580   ;
                          20581   ;
                          20582   ;   create data block of zero
                          20583   ;
     FC86   E8            20584           inx             ;  .x=0
     FC87   8A            20585           txa
                          20586   ;
     FC88   9D 0500       20587   crtdat  sta  buff2,x
     FC8B   E8            20588           inx
     FC8C   D0 FA         20589           bne  crtdat
                          20590   ;
                          20591   ;
                          20592   ;  convert header block to gcr
                          20593   ;
     FC8E   A9 03         20594           lda  #>buff0
     FC90   85 31         20595           sta  bufpnt+1   ;  point at buffer
```

```
Error Addr  Code           Seq    Source statement

                          20596   ;
     FC92   20 FE30       20597           jsr  fbtog      ;  convert to gcr with no bid char
                          20598
     FC95   68            20599           pla             ;   restore block size
     FC96   A8            20600           tay             ;  move buffer up 79 bytes
     FC97   88            20601           dey             ;  for i=n-1 to 0:mem i+69 :=mem i :next
     FC98   20 FDE5       20602           jsr  movup      ;  move buf0 up 69 bytes
                          20603   ;
     FC9B   20 FDF5       20604           jsr  movovr     ;  move ovrbuf up to buffer
                          20605   ;
                          20606   ;
                          20607   ;
                          20608   ;   convert data block to gcr
                          20609   ;   write image
                          20610   ;
                          20611   ;   leave it in ovrbuf and buffer
                          20612   ;
                          20613   ;
     FC9E   A9 05         20614           lda  #>buff2    ;  point at buffer
     FCA0   85 31         20615           sta  bufpnt+1
                          20616   ;
                          20617   ;
     FCA2   20 F5E9       20618           jsr  chkblk     ;  get block checksum
     FCA5   85 3A         20619           sta  chksum
     FCA7   20 F78F       20620           jsr  bingcr
                          20621   ;
                          20622   ;
                          20623   ;
                          20624   ;   start the format now
                          20625   ;
                          20626   ;   write out sync header gap1
                          20627   ;   data block
                          20628   ;
                          20629   ;
                          20630   ;
     FCAA   A9 00         20631           lda  #0          ;  init counter
     FCAC   85 32         20632           sta  hdrpnt
                          20633   ;
     FCAE   20 FE0E       20634           jsr  clear      ;  clear disk
                          20635   ;
     FCB1   A9 FF         20636   wrtsyn  lda  #$ff       ;  write sync
     FCB3   8D 1C01       20637           sta  data2
                          20638   ;
     FCB6   A2 05         20639           ldx  #numsyn    ;  write 4  sync
                          20640   ;
     FCB8   50 FE         20641   wrts10  bvc  *
     FCBA   B8            20642           clv
                          20643   ;
     FCBB   CA            20644           dex
     FCBC   D0 FA         20645           bne  wrts10
```

```
Error Addr  Code           Seq    Source statement

                          20646   ;
     FCBE   A2 0A         20647           ldx  #10        ;  write out header
     FCC0   A4 32         20648           ldy  hdrpnt
                          20649   ;
     FCC2   50 FE         20650   wrts20  bvc  *
     FCC4   B8            20651           clv
                          20652   ;
     FCC5   B9 0300       20653           lda  buff0,y    ;  get header data
     FCC8   8D 1C01       20654           sta  data2
                          20655   ;
     FCCB   C8            20656           iny
     FCCC   CA            20657           dex
     FCCD   D0 F3         20658           bne  wrts20
                          20659   ;
                          20660   ;
                          20661   ; * write out gap1
                          20662   ;
     FCCF   A2 09         20663           ldx  #gap1-2    ;  write  gcr bytes
                          20664   ;
     FCD1   50 FE         20665   wrts30  bvc  *
     FCD3   B8            20666           clv
                          20667   ;
     FCD4   A9 55         20668           lda  #$55
     FCD6   8D 1C01       20669           sta  data2
                          20670   ;
     FCD9   CA            20671           dex
     FCDA   D0 F5         20672           bne  wrts30
                          20673   ;
                          20674   ;
                          20675   ;
                          20676   ; * write out data block
                          20677   ;
     FCDC   A9 FF         20678           lda  #$ff       ;  write data block sync
                          20679   ;
     FCDE   A2 05         20680           ldx  #numsyn
                          20681   ;
     FCE0   50 FE         20682   dbsync  bvc  *
     FCE2   B8            20683           clv
                          20684   ;
     FCE3   8D 1C01       20685           sta  data2
                          20686   ;
     FCE6   CA            20687           dex
     FCE7   D0 F7         20688           bne  dbsync
                          20689   ;
     FCE9   A2 BB         20690           ldx  #256-topwrt ;  write out ovrbuf
                          20691   ;
     FCEB   50 FE         20692   wrts40  bvc  *
     FCED   B8            20693           clv
                          20694   ;
     FCEE   BD 0100       20695           lda  ovrbuf,x
```

```
Error Addr  Code           Seq    Source statement

     FCF1   8D 1C01      20696            sta  data2
                         20697    ;
     FCF4   E8           20698            inx
     FCF5   D0 F4        20699            bne  wrts40
                         20700    ;
                         20701    ;
     FCF7   A0 00        20702            ldy  #0
                         20703    ;
     FCF9   50 FE        20704    wrts50  bvc  *
     FCFB   B8           20705            clv
                         20706    ;
     FCFC   B1 30        20707            lda  (bufpnt),y
     FCFE   8D 1C01      20708            sta  data2
                         20709    ;
     FD01   C8           20710            iny
     FD02   D0 F5        20711            bne  wrts50
                         20712    ;
     FD04   A9 55        20713            lda  #$55        ;  write gap2(dtrck)
     FD06   AE 0626      20714            ldx  dtrck
                         20715    ;
     FD09   50 FE        20716    wgp2    bvc  *
     FD0B   B8           20717            clv
                         20718    ;
     FD0C   8D 1C01      20719            sta  data2
     FD0F   CA           20720            dex
     FD10   D0 F7        20721            bne  wgp2
                         20722    ;
                         20723    ;       ldx #20          ; write erase trail gap
                         20724    ;wgp3   bvc *
                         20725    ;       clv
                         20726    ;       dex
                         20727    ;       bne wgp3
                         20728    ;
     FD12   A5 32        20729            lda  hdrpnt      ;  advance header pointer
     FD14   18           20730            clc
     FD15   69 0A        20731            adc  #10
     FD17   85 32        20732            sta  hdrpnt
                         20733    ;
                         20734    ;
                         20735    ;
                         20736    ;  done writing sector
                         20737    ;
     FD19   CE 0628      20738            dec  sect        ;  go to next on
     FD1C   D0 93        20739            bne  wrtsyn      ;  more to do
                         20740    ;
     FD1E   50 FE        20741            bvc  *           ;  wait for last one to write
     FD20   B8           20742            clv
                         20743    ;
     FD21   50 FE        20744            bvc  *
     FD23   B8           20745            clv
```

```
Error Addr  Code           Seq    Source statement

                          20746   ;
     FD24   20 FE00       20747           jsr  kill       ;  goto read mode
                          20748   ;
                          20749   ;
                          20750   ;
                          20751   ;.end
                          20752           .include  lccfmt3
```

```
Error Addr  Code           Seq    Source statement

                          20754            .subttl 'lccfmt3.src'
                          20755    ;
                          20756    ;
                          20757    ;
                          20758    ;*   format done, now verify it
                          20759    ;
                          20760    ;
                          20761    ;
     FD27   A9 C8         20762            lda  #200       ;  look at 200 syncs
     FD29   8D 0623       20763            sta  trys
                          20764    ;
     FD2C   A9 00         20765    comp    lda  #0         ;  pointer into headers
     FD2E   85 30         20766            sta  bufpnt
                          20767
     FD30   A9 03         20768            lda  #>buff0
     FD32   85 31         20769            sta  bufpnt+1
                          20770
     FD34   A5 43         20771            lda  sectr      ;  sector counter
     FD36   8D 0628       20772            sta  sect
                          20773    ;
     FD39   20 F556       20774    cmpr10  jsr  sync       ;  find sync
                          20775    ;
     FD3C   A2 0A         20776            ldx  #10
     FD3E   A0 00         20777            ldy  #0
                          20778    ;
     FD40   50 FE         20779    cmpr15  bvc  *          ;  get header byte
     FD42   B8            20780            clv
                          20781    ;
     FD43   AD 1C01       20782            lda  data2
     FD46   D1 30         20783            cmp  (bufpnt),y ;  compare gcr
                          20784    ;
     FD48   D0 0E         20785            bne  cmpr20     ; error
                          20786    ;
     FD4A   C8            20787            iny
     FD4B   CA            20788            dex
     FD4C   D0 F2         20789            bne  cmpr15     ;  test all bytes
                          20790    ;
     FD4E   18            20791            clc             ;  update headr pointer
     FD4F   A5 30         20792            lda  bufpnt
     FD51   69 0A         20793            adc  #10
     FD53   85 30         20794            sta  bufpnt
                          20795    ;
     FD55   4C FD62       20796            jmp  tstdat     ;  now test data
                          20797    ;
     FD58   CE 0623       20798    cmpr20  dec  trys       ;  test if too many errors
     FD5B   D0 CF         20799            bne  comp
                          20800    ;
     FD5D   A9 06         20801            lda  #notfnd    ;  too many error
     FD5F   4C FDD3       20802            jmp  fmterr
                          20803
```

```
Error Addr  Code          Seq    Source statement

     FD62   20 F556      20804   tstdat  jsr  sync        ;  find data sync
                         20805   ;
     FD65   A0 BB        20806           ldy  #256-topwrt
                         20807   ;
     FD67   50 FE        20808   tst05   bvc  *
     FD69   B8           20809           clv
                         20810   ;
     FD6A   AD 1C01      20811           lda  data2       ;  compare gcr
     FD6D   D9 0100      20812           cmp  ovrbuf,y
                         20813   ;
     FD70   D0 E6        20814           bne  cmpr20      ;  error
                         20815   ;
     FD72   C8           20816           iny
     FD73   D0 F2        20817           bne  tst05       ;  do all ovrbuf
                         20818   ;
     FD75   A2 FC        20819           ldx  #255-3      ;  now do buffer, dont test off bytes
                         20820   ;
     FD77   50 FE        20821   tst10   bvc  *
     FD79   B8           20822           clv
                         20823   ;
     FD7A   AD 1C01      20824           lda  data2
     FD7D   D9 0500      20825           cmp  buff2,y
     FD80   D0 D6        20826           bne  cmpr20
                         20827   ;
     FD82   C8           20828           iny
     FD83   CA           20829           dex
     FD84   D0 F1        20830           bne  tst10
                         20831   ;
                         20832   ;
     FD86   CE 0628      20833           dec  sect        ;  more sectors to test?
     FD89   D0 AE        20834           bne  cmpr10      ;  yes
                         20835   ;
                         20836   ;
                         20837   ;  all sectors done ok
                         20838   ;
     FD8B   E6 51        20839           inc  ftnum       ;  goto next track
     FD8D   A5 51        20840           lda  ftnum
     FD8F   C9 24        20841           cmp  #36         ;  #tracks max
     FD91   B0 03        20842           bcs  fmtend
                         20843   ;
     FD93   4C F99C      20844           jmp  end         ;  more to do
                         20845   ;
                         20846   ;
     FD96   A9 FF        20847   fmtend  lda  #$ff        ;  clear ftnum
     FD98   85 51        20848           sta  ftnum
                         20849   ;
     FD9A   A9 00        20850           lda  #$0         ;  clear gcr buffer flag
     FD9C   85 50        20851           sta  gcrflg
                         20852   ;
     FD9E   A9 01        20853           lda  #1          ;  return ok code
```

```
Error Addr  Code          Seq   Source statement

     FDAO   4C F969     20854           jmp  errr
                        20855   ;
                        20856   ;
                        20857   ;
                        20858   ;.end
                        20859           .include  lccfmt4
```

```
Error Addr  Code           Seq    Source statement

                          20861            .subttl 'lccfmt4.src'
                          20862    ;
                          20863    ;
                          20864    ;
     FDA3   AD 1C0C       20865    synclr  lda  pcr2        ;  write entire track with sync
     FDA6   29 1F         20866            and  #$ff-$e0
     FDA8   09 C0         20867            ora  #$c0
     FDAA   8D 1C0C       20868            sta  pcr2
                          20869    ;
     FDAD   A9 FF         20870            lda  #$ff        ;  output mode ddr
     FDAF   8D 1C03       20871            sta  ddra2
                          20872    ;
     FDB2   8D 1C01       20873            sta  data2       ;  sync char
                          20874    ;
     FDB5   A2 28         20875            ldx  #$28        ;  $28*256 bytes
     FDB7   A0 00         20876            ldy  #0
                          20877    ;
     FDB9   50 FE         20878    syc10   bvc  *
     FDBB   B8            20879            clv
                          20880    ;
     FDBC   88            20881            dey
     FDBD   D0 FA         20882            bne  syc10
                          20883    ;
     FDBF   CA            20884            dex
     FDC0   D0 F7         20885            bne  syc10
                          20886    ;
     FDC2   60            20887            rts              ;  leave write on
                          20888    ;
                          20889    ;
                          20890    ;
     FDC3   AE 0621       20891    wrtnum  ldx  num         ;  write out num bytes
     FDC6   AC 0622       20892            ldy  num+1
                          20893    ;
     FDC9   50 FE         20894    wrtn10  bvc  *
     FDCB   B8            20895            clv
                          20896    ;
     FDCC   CA            20897            dex
     FDCD   D0 FA         20898            bne  wrtn10
                          20899    ;
     FDCF   88            20900            dey
     FDD0   10 F7         20901            bpl  wrtn10
                          20902    ;
     FDD2   60            20903            rts
                          20904    ;
                          20905    ;
                          20906    ;
     FDD3   CE 0620       20907    fmterr  dec  cnt         ;  test for retry
     FDD6   F0 03         20908            beq  fmte10
                          20909    ;
     FDD8   4C F99C       20910            jmp  end
```

```
Error Addr  Code           Seq    Source statement

                          20911   ;
     FDDB                 20912   fmte10
                          20913   ;
     FDDB   A0 FF         20914           ldy  #$ff
     FDDD   84 51         20915           sty  ftnum      ;  clear format
                          20916   ;
     FDDF   C8            20917           iny
     FDE0   84 50         20918           sty  gcrflg
                          20919   ;
     FDE2   4C F969       20920           jmp  errr
                          20921   ;
                          20922   ;
                          20923   ;
     FDE5   B9 0300       20924   movup   lda  buff0,y    ;   move up 69 bytes
     FDE8   99 0345       20925           sta  buff0+69,y ;  move from top down
     FDEB   88            20926           dey
     FDEC   D0 F7         20927           bne  movup
                          20928
     FDEE   AD 0300       20929           lda  buff0      ;  do last byte
     FDF1   8D 0345       20930           sta  buff0+69
     FDF4   60            20931           rts
                          20932   ;
                          20933   ;
     FDF5   A0 44         20934   movovr  ldy  #68        ;  move ovrbuf into (buffer)
                          20935   ;
     FDF7   B9 01BB       20936   movo10  lda  ovrbuf+256-topwrt,y
     FDFA   91 30         20937           sta  (bufpnt),y
                          20938   ;
     FDFC   88            20939           dey
     FDFD   10 F8         20940           bpl  movo10
                          20941   ;
     FDFF   60            20942           rts
                          20943   ;
                          20944   ;
                          20945   ;
     FE00   AD 1C0C       20946   kill    lda  pcr2       ;  disable write
     FE03   09 E0         20947           ora  #$e0
     FE05   8D 1C0C       20948           sta  pcr2
     FE08   A9 00         20949           lda  #$00       ;  make port input now
     FE0A   8D 1C03       20950           sta  ddra2
                          20951   ;
     FE0D   60            20952           rts
                          20953   ;
                          20954   ;
                          20955   ;
     FE0E   AD 1C0C       20956   clear   lda  pcr2       ;  enable write
     FE11   29 1F         20957           and  #$ff-$e0
     FE13   09 C0         20958           ora  #$c0
     FE15   8D 1C0C       20959           sta  pcr2
                          20960   ;
```

```
Error Addr  Code           Seq    Source statement

     FE18   A9 FF        20961            lda  #$ff        ;  make port an output
     FE1A   8D 1C03      20962            sta  ddra2
                         20963    ;
     FE1D   A9 55        20964            lda  #$55        ;  write a 1f pattern
     FE1F   8D 1C01      20965            sta  data2
                         20966    ;
     FE22   A2 28        20967            ldx  #$28        ;  $28*256 chars
     FE24   A0 00        20968            ldy  #00
     FE26   50 FE        20969    cler10  bvc  *
     FE28   B8           20970            clv
     FE29   88           20971            dey
     FE2A   D0 FA        20972            bne  cler10
                         20973    ;
     FE2C   CA           20974            dex
     FE2D   D0 F7        20975            bne  cler10
                         20976    ;
     FE2F   60           20977            rts
                         20978    ;
                         20979    ;*****************************
                         20980    ;*
                         20981    ;*
                         20982    ;*      fbtog
                         20983    ;*      format binary to gcr conversion
                         20984    ;*
                         20985    ;*      converts buffer to gcr with out hbid
                         20986    ;*
                         20987    ;*************************
                         20988    ;
     FE30   A9 00        20989    fbtog   lda  #0          ;  point at buffer
     FE32   85 30        20990            sta  bufpnt
     FE34   85 2E        20991            sta  savpnt
     FE36   85 36        20992            sta  bytcnt
                         20993    ;
     FE38   A9 BB        20994            lda  #256-topwrt ;  put gcr in ovrflow buffer
     FE3A   85 34        20995            sta  gcrpnt
                         20996    ;
     FE3C   A5 31        20997            lda  bufpnt+1    ;  save buffer pointer
     FE3E   85 2F        20998            sta  savpnt+1
                         20999    ;
     FE40   A9 01        21000            lda  #>ovrbuf
     FE42   85 31        21001            sta  bufpnt+1    ;  store in overbuf
                         21002    ;
     FE44   A4 36        21003    fbg10   ldy  bytcnt      ;  get pointer
                         21004    ;
     FE46   B1 2E        21005            lda  (savpnt),y
     FE48   85 52        21006            sta  btab
     FE4A   C8           21007            iny
                         21008    ;
     FE4B   B1 2E        21009            lda  (savpnt),y
     FE4D   85 53        21010            sta  btab+1
```

```
Error Addr  Code           Seq    Source statement

     FE4F   C8           21011            iny
                         21012    ;
     FE50   B1 2E        21013            lda  (savpnt),y
     FE52   85 54        21014            sta  btab+2
     FE54   C8           21015            iny
                         21016    ;
     FE55   B1 2E        21017            lda  (savpnt),y
     FE57   85 55        21018            sta  btab+3
     FE59   C8           21019            iny
     FE5A   F0 08        21020            beq  fbg15       ;  test if done
                         21021    ;
     FE5C   84 36        21022            sty  bytcnt      ;  save pointer
                         21023    ;
     FE5E   20 F6D0      21024            jsr  put4bg      ;  convert and store
                         21025    ;
     FE61   4C FE44      21026            jmp  fbg10
                         21027    ;
     FE64   4C F6D0      21028    fbg15   jmp  put4bg      ;  done, return
                         21029    ;
                         21030    ;.end
                         21031            .include  irq
```

```
Error Addr  Code           Seq    Source statement

                          21033            .subttl 'irq.src'
                          21034    ;<><><>
```

```
Error Addr  Code           Seq    Source statement

                          21067           .subttl 'romtblsf.src'
                          21068
     FE85   12            21069   dirtrk   .byte  18       ; directory track #
     FE86   04            21070   bamsiz   .byte  4        ; # bytes/track in bam
     FE87   04            21071   mapoff   .byte  4        ; offset of bam in sector
     FE88   90            21072   dsknam   .byte  $90      ; offset of disk name in bam sector
                          21073   ;
                          21074   ;   command search table
     FE89   56 49 44      21075   cmdtbl   .byte  'VIDMBUP&CRSN'
     FE8C   4D 42 55
     FE8F   50 26 43
     FE92   52 53 4E
                          21076   ; validate-dir init-drive duplicate
                          21077   ; memory-op block-op user
                          21078   ; position dskcpy utlodr rename scratch new
          =000C           21079   ncmds    =*-cmdtbl
                          21080   ;  jump table low
     FE95   84 05 C1      21081   cjumpl   .byte    <verdir,<intdrv,<duplct
     FE98   F8 1B 5C      21082            .byte    <mem,<block,<user
     FE9B   07            21083            .byte    <record
     FE9C   A3            21084            .byte    <utlodr
     FE9D   F0            21085            .byte    <dskcpy
     FE9E   88 23 0D      21086            .byte    <rename,<scrtch,<new
          =FEA1           21087   *=cjumpl+ncmds
                          21088   ;  jump table high
     FEA1   ED D0 C8      21089   cjumph   .byte    >verdir,>intdrv,>duplct
     FEA4   CA CC CB      21090            .byte    >mem,>block,>user
     FEA7   E2            21091            .byte    >record
     FEA8   E7            21092            .byte    >utlodr
     FEA9   C8            21093            .byte    >dskcpy
     FEAA   CA C8 EE      21094            .byte    >rename,>scrtch,>new
          =FEAD           21095   *=cjumph+ncmds
          =0000           21096   val=0            ; validate (verify) cmd #
                          21097
                          21098   ; structure images for cmds
          =0009           21099   pcmd     =9
     FEAD   51            21100           .byte    %01010001       ;  dskcpy
          =FEA5           21101   struct   =*-pcmd                 ;  cmds not parsed
     FEAE   DD            21102           .byte    %11011101       ;  rename
     FEAF   1C            21103           .byte    %00011100       ;  scratch
     FEB0   9E            21104           .byte    %10011110       ;  new
          =000C           21105   ldcmd    =*-struct               ;  load cmd image
     FEB1   1C            21106           .byte    %00011100       ;  load
                          21107   ;            --- ---
                          21108   ;           pgdrpgdr
                          21109   ;           fs1 fs2
                          21110
                          21111   ;   bit reps:  not pattern
                          21112   ;              not greater than one file
                          21113   ;              not default drive(s)
```

```
Error Addr  Code           Seq    Source statement

                          21114   ;               required filename
                          21115
     FEB2   52 57 41      21116   modlst    .byte  'RWAM'           ;  mode table
     FEB5   4D
            =0004         21117   nmodes    =*-modlst
                          21118   ;file type table
     FEB6   44 53 50      21119   tplst     .byte  'DSPUL'
     FEB9   55 4C
     FEBB   44 53 50      21120   typlst    .byte  'DSPUR'           ; del, seq, prog, user, relative
     FEBE   55 52
            =0005         21121   ntypes    =*-typlst
     FEC0   45 45 52      21122   tp1lst    .byte  'EERSE'
     FEC3   53 45
     FEC5   4C 51 47      21123   tp2lst    .byte  'LQGRL'
     FEC8   52 4C
     FECA   08 00         21124   ledmsk    .byte  led0,led1
                          21125   ;
                          21126   ; error flag vars for bit
                          21127   ;
     FECC   00            21128   er00      .byte  0
     FECD   3F            21129   er0       .byte  $3f
     FECE   7F            21130   er1       .byte  $7f
     FECF   BF            21131   er2       .byte  $bf
     FED0   FF            21132   er3       .byte  $ff
                          21133   ;
     FED1                 21134   numsec                             ; (4) sectors/track
     FED1   11 12 13      21135             .byte    17,18,19,21
     FED4   15
     FED5   41            21136   vernum    .byte  fm4040            ; format type
     FED6   04            21137   nzones    .byte  4                 ; # of zones
                          21138
                          21139   ; maxtrk                           ; maximum track #  +1
     FED7                 21140   trknum                             ; zone boundaries track numbers
     FED7   24 1F 19      21141             .byte    36,31,25,18
     FEDA   12
     FEDB                 21142   offset                             ; for recovery
     FEDB   01 FF FF      21143             .byte    1,$ff,$ff,1,0
     FEDE   01 00
                          21144   ;
     FEE0                 21145   bufind
     FEE0   03 04 05      21146             .byte    $03,$04,$05,$06,$07,$07
     FEE3   06 07 07
                          21147   ;
                          21148
                          21149   ;.end
                          21150           .include  vector
```

```
Error Addr  Code           Seq    Source statement

                          21152            .subttl 'vector.src'
                          21153
     FEE6   FF            21154    echksm  .byte  $ff       ; << to be defined <<
     FEE7   6C 0065       21155    nmi     jmp  (vnmi)
                          21156    ;
                          21157    ;---------------------------------------------------
                          21158    ;
     FEEA   8D 1C00       21159    pea7a   sta  ledprt      ; patch for power-on errors
     FEED   8D 1C02       21160            sta  ledout      ; turn ddrb to output
     FEF0   4C EA7D       21161            jmp  rea7d       ; return to led blink code
                          21162    ;
                          21163    ;---------------------------------------------------
                          21164    ; patch area for 1541 disk with slow serial receive
                          21165    ;
     FEF3   8A            21166    slowd   txa              ; only affect .a
     FEF4   A2 05         21167            ldx  #5          ; insert 40us of delay with this routine
     FEF6   CA            21168    slowe   dex
     FEF7   D0 FD         21169            bne  slowe
     FEF9   AA            21170            tax
     FEFA   60            21171            rts
                          21172    ;
                          21173    ;---------------------------------------------------
                          21174    ;
     FEFB   20 E9AE       21175    clkdat  jsr  clklow      ; new clock low and...
     FEFE   4C E99C       21176            jmp  dathi       ; data hi for patch area
                          21177    ;
                          21178    ;---------------------------------------------------
                          21179    ;
     FF01   AD 0202       21180    nnmi    lda  cmdbuf+2    ; new nmi routine check for
     FF04   C9 2D         21181            cmp  #'-'
     FF06   F0 05         21182            beq  nnmi10      ; if ui- then no delay
     FF08   38            21183            sec
     FF09   E9 2B         21184            sbc  #'+'
     FF0B   D0 DA         21185            bne  nmi         ; if not ui+ then must be a real ui command
     FF0D   85 23         21186    nnmi10  sta  drvtrk+1
     FF0F   60            21187            rts
                          21188    ;
                          21189    ;
                          21190    ;---------------------------------------------------
                          21191    ;       patch 5   *rom-05 8/18/83*
                          21192    ;
                          21193    ;clock line hi on pwr on
                          21194    ;
     FF10   8E 1803       21195    patch5  stx  ddra1       ; set direction
     FF13   A9 02         21196            lda  #$02        ; set clock high
                          21197
     FF15   4C A65A       21198            jmp  ptch22      ; setup port a
                          21199    ;       sta  pb
                          21200
     FF18   A9 1A         21201    ptch22r lda  #$1a        ; set ddra reg b
```

```
Error Addr  Code           Seq    Source statement

     FF1A   8D 1802      21202            sta  ddrb1
     FF1D   4C EAA7      21203            jmp  dkit10      ; return
                         21204    ;
                         21205    ;
                         21206    ;-------------------------------------------------
                         21207    ;      patch 6   *rom ds 8/18/83*
                         21208    ;
                         21209    ;fix eoi timing to wait all devices
                         21210    ;
                         21211    ;
     FF20   AD 1800      21212    patch6  lda  pb          ; test data line
     FF23   29 01        21213            and  #$01        ; every one rdy ?
     FF25   D0 F9        21214            bne  patch6
     FF27   A9 01        21215            lda  #$01        ; wait 255 usec
     FF29   8D 1805      21216            sta  t1hc1       ; set timer
     FF2C   4C E9DF      21217            jmp  acp00       ; return
                         21218    ;
                         21219    ;
                         21220    ;-------------------------------------------------
                         21221    ;     patch 7   *** rom ds 03/15/85 ***
                         21222    ;
                         21223    ;
     FF2F   A9 FF        21224    patch7  lda  #$ff        ; clear format flags
     FF31   85 51        21225            sta  ftnum       ;
                         21226
     FF33   AD 180F      21227            lda  pota1       ; 1/2 Mhz ?
     FF36   29 20        21228            and  #$20
     FF38   D0 03        21229            bne  1$
                         21230
     FF3A   A9 24        21231            lda  #36
     FF3C   2C           21232            .byte skip2
     FF3D   A9 47        21233    1$      lda  #71
     FF3F   8D 02AC      21234            sta  maxtrk
     FF42   4C A779      21235            jmp  ptch28      ; ***rom ds 02/01/85***
                         21236    ;       jmp  format      ; transfer format to ram
                         21237
                         21238    ;
                         21239    ;
                         21240    ;-------------------------------------------------
                         21241    ;
                         21242    ;     patch 9   *rom ds 09/12/84*
                         21243    ;
     FF45   98           21244    patch9  tya
     FF46   48           21245            pha
     FF47   A0 64        21246    1$      ldy  #100        ; timer (100*16us=1.6ms)
     FF49   AD 180F      21247    2$      lda  pota1       ; check photo
     FF4C   6A           21248            ror  a           ; set/clear carry
     FF4D   08           21249            php              ; save status
     FF4E   AD 180F      21250            lda  pota1
     FF51   6A           21251            ror  a           ; => carry
```

```
Error Addr  Code           Seq    Source statement

     FF52   6A           21252            ror  a           ; => bit 7
     FF53   28           21253            plp              ; get carry
     FF54   29 80        21254            and  #$80        ; set/clear sign bit
     FF56   90 04        21255            bcc  5$
                         21256
     FF58   10 1D        21257            bpl  4$          ; carry set(off) & sign(on) exit
                         21258
     FF5A   30 02        21259            bmi  6$          ; bra
                         21260
     FF5C   30 19        21261   5$       bmi  4$          ; carry clear(on) & sign(off) exit
                         21262
     FF5E   88           21263   6$       dey
     FF5F   D0 E8        21264            bne  2$          ; wait
                         21265
     FF61   B0 14        21266   3$       bcs  4$          ; no, not yet
                         21267
     FF63   AD 1C00      21268            lda  dskcnt      ; yes, check motor phase
     FF66   29 03        21269            and  #$03        ; phase a ?
     FF68   D0 0D        21270            bne  4$          ; no, not yet
                         21271
     FF6A   A5 7B        21272            lda  adrsed      ; chk micro-stepping flag
     FF6C   D0 09        21273            bne  4$
                         21274
     FF6E   68           21275            pla
     FF6F   A8           21276            tay
     FF70   A9 00        21277            lda  #$00        ; yes, clear # of steps
     FF72   85 4A        21278            sta  steps       ;
     FF74   4C FABE      21279            jmp  end33       ; go back to end
                         21280
     FF77   68           21281   4$       pla
     FF78   A8           21282            tay
                         21283
     FF79   E6 4A        21284            inc  steps       ; update # of steps
                         21285
     FF7B   AE 1C00      21286            ldx  dskcnt      ; update motor phase
     FF7E   CA           21287            dex              ;
     FF7F   4C FA38      21288            jmp  pppppp      ; return to main
                         21289   ;
                         21290   ;
                         21291   ;-------------------------------------------------
                         21292   ;
                         21293   ;     patch 10  *rom ds 01/22/85*
                         21294   ;
     FF82   20 F259      21295   ptch10   jsr  cntint     ; controller init.
                         21296
     FF85   A9 05        21297            lda  #5
     FF87   85 3C        21298            sta  cpmit       ; set interleave for CP/M
     FF89   A9 88        21299            lda  #<irq
     FF8B   8D 02A9      21300            sta  irqjmp
     FF8E   A9 9D        21301            lda  #>irq
```

```
Error Addr  Code           Seq    Source statement

     FF90   8D 02AA      21302            sta  irqjmp+1   ; vector 1541
                         21303
     FF93   A9 24        21304            lda  #36
     FF95   8D 02AC      21305            sta  maxtrk     ; set maximum track+1
                         21306
     FF98   18           21307            clc             ; set job header
     FF99   4C 93F3      21308            jmp  set_side   ; side zero
                         21309   ;
                         21310   ;
                         21311   ;-----------------------------------------------------
                         21312   ;     patch 11  *rom ds 01/21/85*
                         21313   ;
     FF9C   85 FF        21314   ptch11  sta  nodrv       ; clr nodrv
     FF9E   4C C100      21315            jmp  setlds     ; set leds
                         21316   ;
                         21317   ;
                         21318   ;-----------------------------------------------------
                         21319   ;     patch 12  *rom ds 01/21/85*
                         21320   ;
     FFA1   85 7B        21321   ptch12  sta  adrsed      ; set micro-stepping flag
     FFA3   4C D676      21322            jmp  hedoff     ; move head now
                         21323   ;
                         21324   ;
                         21325   ;-----------------------------------------------------
                         21326   ;     patch 13  *rom ds 01/21/85*
                         21327   ;
     FFA6   20 D676      21328   ptch13  jsr  hedoff      ; restore head
     FFA9   A9 00        21329            lda  #$00
     FFAB   85 7B        21330            sta  adrsed     ; clear micro-stepping flag
     FFAD   60           21331            rts
                         21332   ;
                         21333   ;
                         21334   ;-----------------------------------------------------
                         21335   ;     patch 14
                         21336   ;
                         21337   ; ////// POST NO BILLS ///////
                         21338   ;
                         21339   ;-----------------------------------------------------
                         21340   ;     patch 15  *rom ds 01/21/85*
                         21341   ;
     FFAE   A4 82        21342   ptch15  ldy  lindx
     FFB0   4C D3DE      21343            jmp  rndget
                         21344   ;
                         21345   ;
                         21346   ;-----------------------------------------------------
                         21347   ;
                         21348   ;default table for user command
                         21349   ;
            =FFE6        21350   *=       $ffe6
     FFE6   C8C6         21351            .word    format
```

```
Error Addr  Code           Seq    Source statement

     FFE8   F98F         21352            .word    trnoff
     FFEA   CD5F         21353   ublock   .word    ublkrd
     FFEC   CD97         21354            .word    ublkwt
                         21355
     FFEE   0500         21356            .word    $0500          ; links to buffer #2
     FFF0   0503         21357            .word    $0503
     FFF2   0506         21358            .word    $0506
     FFF4   0509         21359            .word    $0509
     FFF6   050C         21360            .word    $050c
     FFF8   050F         21361            .word    $050f
            =FFFA        21362   *=        $fffa
     FFFA   FF01         21363            .word    nnmi           ; goto new nmi routine
     FFFC   EAA0         21364            .word    dskint
     FFFE   FE67         21365            .word    sysirq
                         21366            .subttl 'The End ...'
                         21367            .end

     No errors detected
```

* * Symbol Table * *

A10	813A	A11	818D	A12	8177	A13	818A
A14	8171	A2	815A	A3	80FA	A4	810D
A5	811E	A6	811B	A7	812C	A8	8155
A9	8192	ABRSK	8A65	AC10	F22B	AC20	F22D
AC30	F236	AC40	F246	ACC200	CEE6	ACCUM	008F
ACCX2	CEE5	ACCX4	CEE2	ACLSTP	0060	ACLTIM	0048
ACLTIM2	0035	ACP00	E9DF	ACP00A	E9CD	ACP00B	E9F2
ACP01	EA0B	ACP02	E9F7	ACP02A	E9FD	ACP03	EA0B
ACP03A	EA1A	ACPTR	E9C9	ACR1	180B	ACR2	1C0B
ACTJOB	001B	AD10	C325	ADD100	CEF0	ADDFIL	D6E4
ADDNR	E304	ADDR1	E33B	ADDREL	E31C	ADDRES	CEED
ADDRTS	DF65	ADDSUM	E84B	ADDT12	DF5C	ADRSED	007B
AF	005F	AF08	D715	AF10	D726	AF15	D730
AF20	D73D	AF25	D74D	AF30	D76F	AF50	D790
AH10	CCAB	AH20	CCCA	AH30	CCD0	AH35	CCD7
AH40	CCE4	ALLDRS	C320	ALLOC_B	A51E	AN05	E318
AN10	E31B	AP30	DA42	APMODE	=0002	APPEND	DA2A
AR10	E355	AR20	E363	AR25	E368	AR30	E372
AR35	E38F	AR40	E39D	AR45	E3B6	AR50	E3C8
AR55	E3D4	AR60	E3F9	AR65	E418	AR70	E444
AS	005E	ASCHEX	CCA1	ATN	=0080	ATN100	E8ED
ATN110	E8FA	ATN120	E8FD	ATN122	E902	ATN30	E884
ATN35	E891	ATN40	E89B	ATN45	E8A9	ATN50	E8B7
ATN95	E8D2	ATNA	=0010	ATNIRQ	E853	ATNLOW	EA56
ATNMOD	007D	ATNPND	007C	ATNS15	E87B	ATNS20	E8D7
ATNSRV	E85B	AUTO1	C65F	AUTO2	C669	AUTOFG	0068
AUTOI	C63D	AVCK	F220	B02	DEB9	BOTOBO	DEA5
B2X10	F118	BA10	CD06	BA15	CD19	BA20	CD1A
BA30	CD2C	BA40	CD31	BADCMD	=0031	BADFN	=0033
BADID	F41B	BADSYN	=0030	BADTS	=0066	BAM	02A1
BAM2A	F10F	BAM2X	F119	BAMSIZ	FE86	BAM_ONE	0146
BAM_PT	A4E7	BCD2	E6B4	BCDDEC	E6AB	BCJMP	CC63
BCTAB	CC5D	BD_INT	88BE	BE05	CDAB	BE10	CDBA
BFCNT	=0005	BFONLY	83D0	BGTAB	F77F	BH10	E2DC
BHERE	E2D0	BHERE2	E2D3	BID	0038	BIGFIL	=0052
BINGO7	F7BA	BING20	F7D9	BING35	F76F	BINGCR	F78F
BITCNT	0037	BKOTST	CDF2	BLINDX	=0006	BLK10	CC26
BLK30	CC2B	BLK40	CC30	BLK50	CC38	BLK60	CC42
BLKALC	CD03	BLKEXC	CDA3	BLKFRE	CCF5	BLKNB	C7AC
BLKNB1	C7B0	BLKPAR	CC6F	BLKPTR	CDBD	BLKRD	CD56
BLKRD2	CD36	BLKRD3	CD42	BLKTST	CDF5	BLKWT	CD73
BLOCK	CC1B	BMASK	EFE9	BMINDX	A5DB	BMP	F37C
BMPNT	006D	BOOT	E780	BOOT2	E77F	BOOT3	E780
BOOT4	E7A8	BP05	CC7C	BP10	CC8B	BP20	CC92
BREAK	E202	BT05	CE08	BT15	CDE0	BT20	CDE5
BTAB	0052	BUF0	00A7	BUF1	00AE	BUFF0	=0300
BUFF1	=0400	BUFF2	=0500	BUFIND	FEE0	BUFPNT	0030
BUFS	=0300	BUFTAB	0099	BUFTST	CDD2	BUFUSE	024F
BUMP	=00C0	BUMPC	=0040	BURST	86A0	BURST_ROUTINES	8030
BURTST	869D	BW10	CD81	BW20	CD8C	BYTCNT	0036

* * Symbol Table * *

BYTE	003C	BYT_CLR	=1C00	CB	=00A3	CB10	DECC
CBMV2	=0073	CBPTR	=000A	CC10	C594	CC15	C59A
CC20	C5A6	CCHKSM	C000	CDIRTY	E115	CDRIVE	003E
CHANNEL	02B0	CHAR	0275	CHCSEE	891B	CHGUTL	8FE5
CHKB10	F5EC	CHKBLK	F5E9	CHKIN	CACC	CHKI0	CAE7
CHKSEE	85AF	CHKSUM	003A	CHK_BLK	A545	CHNDAT	023E
CHNRDY	00F2	CJUMPH	FEA1	CJUMPL	FE95	CK10	CAD6
CK20	CAE6	CK25	CAEA	CK30	CAF4	CKM1	DA11
CKM2	DA1C	CKT1	DA1E	CKT2	DA29	CKTM	DA09
CLB1	F00B	CLD2	D334	CLDCHN	D313	CLEAR	FE0E
CLER10	FE26	CLKDAT	FEFB	CLKHI	E9B7	CLKIN	=0004
CLKLOW	E9AE	CLKOUT	=0008	CLNBAM	F0D1	CLOSE	DAC0
CLR10	E104	CLRB2	C1C1	CLRBAM	F005	CLRBUF	DEC1
CLRC1	D30B	CLRCB	C1BD	CLRCHN	D307	CLREC	E0F3
CLRF10	DDA3	CLRFLG	DD9D	CLS05	DAD1	CLS10	DAD4
CLS15	DAE9	CLS20	DAF0	CLS25	DAFF	CLSALL	DAEC
CLSC28	DB0C	CLSC30	DB26	CLSC31	DB29	CLSCHN	DB02
CLSD	D317	CLSD4	DC06	CLSD5	DC21	CLSD6	DC29
CLSDIR	DBA5	CLSR1	DB5F	CLSREL	DB2C	CLSW10	DB76
CLSW15	DB88	CLSW20	DB8C	CLSWRT	DB62	CMD	024D
CMD10	8E66	CMD100	8E7F	CMD7	8AF3	CMD70	8B09
CMD71	8B1D	CMD72	8B97	CMD73	8BAB	CMD74	8BBF
CMD750	8BE5	CMD75X	8BFE	CMD7A	8BD1	CMD7B	8B86
CMD7C	8B76	CMD7D	8B66	CMD7E	8B4E	CMD7F	8B3E
CMD7FE	8C1A	CMD7FF	8C02	CMD7N	8B2F	CMD9EX	8DF3
CMDBUF	0200	CMDCHN	=0004	CMDCTL	806E	CMDEIG	8C57
CMDELE	8EC6	CMDER2	E645	CMDER3	E648	CMDERR	C1C8
CMDFIV	8A27	CMDFOR	8A08	CMDFVE	8A09	CMDINF	86B9
CMDIS	=001E	CMDJMP	86C8	CMDLEN	=0029	CMDNIN	8D67
CMDNUM	022A	CMDONE	89EF	CMDRET	8FAA	CMDRST	C2DC
CMDSA	=000F	CMDSEC	8FA4	CMDSET	C2B3	CMDSEV	8A86
CMDSIZ	0274	CMDTBB	808E	CMDTBL	FE89	CMDTEN	8DF6
CMDTHI	8F5F	CMDTHR	8A03	CMDTWO	89FD	CMDTWV	8F18
CMDWAT	0255	CMDZER	89EC	CMD_11	8EDC	CMD_12	8F25
CMD_SEC	0096	CMD_TRK	0067	CMPCHK	C589	CMPR10	FD39
CMPR15	FD40	CMPR20	FD58	CNT	=0620	CNT10	FBBB
CNT20	FBCE	CNTINT	F259	CNTST	=F259	CNVBIN	F497
CODE	FAC7	COMP	FD2C	COMPAR	C4D8	CONHDR	F934
CONSEK	8A57	CONT	0098	CONT10	F2C0	CONT20	F2F3
CONT30	F2CD	CONT35	F2D8	CONT40	F2E9	CONT50	F2F0
COPO1	C982	COPO5	C987	COP10	C9A1	COPY	C952
COUNT	FBB6	CPOO	8C90	CPO2	C4E6	CPO5	C4E7
CP10	C4EC	CP20	C4FE	CP30	C50A	CP32	C51B
CP33	C51D	CP34	C52B	CP40	C535	CP42	C55C
CP50	C589	CPMINT	84F1	CPMIT	003C	CPMSEK	0097
CR	=000D	CR20	EAEA	CR30	EAEC	CRA	400E
CRTDAT	FC88	CSO7	C2CA	CSO8	C2CB	CS10	C2FD
CSECT	004C	CSERR	F41E	CTL_CMD	005F	CTL_DAT	0046
CTRACK	0046	CTR_ERR	9199	CURBLK	DE3B	CUR_TRK	0064
CVSTAT	883C	CY	C9A7	CY10	C9B9	CY10A	C9CE

* * Symbol Table * *

CY15	C9D5	CY20	C9D8	CY30	C9EA	CYEXT	CA53
C_100	8E34	C_101	8E04	C_102	8E17	C_103	8E0F
C_104	8E2B	C_105	8EB7	C_106	8EC2	C_107	8EC5
C_108	8E88	C_109	8E9A	C_110	8E93	C_1100	8EE1
C_1101	8F05	C_1102	8EFC	C_1103	8EFE	C_1104	8F10
C_111	8EA0	C_112	8E45	C_1200	8F2A	C_1201	8F4C
C_1202	8F41	C_1203	8F42	C_1204	8F57	C_800	8D0D
C_801	8D5F	C_802	8D40	C_803	8D5C	C_900	8D89
C_901	8DA2	C_902	8DAB	C_904	8DC6	C_905	8DBF
C_906	8DE8	C_907	8DD0	C_908	8DDF	DATA	0085
DATA2	1C01	DATHI	E99C	DATIN	=0001	DATLOW	E9A5
DATOUT	=0002	DBID	0047	DBL05	CF57	DBL08	CF5D
DBL10	CF66	DBL15	CF6C	DBL20	CF6F	DBL30	CF76
DBLBUF	CF1E	DBS10	CF8B	DBSET	CF7B	DBSYNC	FCE0
DDRA1	1803	DDRA2	1C03	DDRB1	1802	DDRB2	1C02
DEALL_B	A508	DEBNC	E9C0	DECTAB	CCF2	DEL1	C8AD
DEL10	C894	DEL2	C894	DELDIR	C8B6	DELFIL	C87D
DELIND	0292	DELSEC	0291	DERR	F1F5	DIAGOK	EB22
DIND	0266	DIR1	ECEA	DIR10	ED0D	DIR3	ED23
DIRBUF	0094	DIRERR	=0071	DIRLEN	=0018	DIRLST	0254
DIRSEC	0290	DIRTRK	FE85	DIRTYP	=0007	DISKIN	89B3
DIV100	CE77	DIV120	CE71	DIV150	CE87	DIV200	CE89
DIV254	CE6E	DIV300	CEA3	DIV400	CEB0	DIV500	CEBF
DIV600	CED6	DIV700	CED8	DKIT10	EAA7	DKMODE	005E
DOIT	D590	DOIT2	D593	DOITT	F473	DOJOB	D58C
DONE	F418	DOREAD	D586	DOREC	D6A6	DOREC1	D6AB
DOREC2	D6B9	DOREC3	D6C4	DOSTEP	FA2E	DOSVER	=0002
DOWRIT	D58A	DO_THE_COMMAND	8761	DRDBYT	D4F6	DRIVE	003D
DRT	D466	DRTRD	D460	DRTWRT	D464	DRVCNT	028C
DRVFLG	028D	DRVMSK	=0001	DRVNUM	007F	DRVST	0020
DRVTRK	0022	DS08	FBE0	DS10	FBEA	DS14	FBF1
DS15	FC03	DS17	FC15	DS20	FC19	DS22	FC21
DS30	FC24	DS32	FC30	DSEC	0260	DSIDE	01AF
DSKCNT	1C00	DSKCPY	C8F0	DSKFUL	=0072	DSKID	0012
DSKIN1	EB76	DSKIN2	EB7E	DSKINT	EAA0	DSKNAM	FE88
DSKVER	0101	DSTRT	F50A	DTRCK	=0626	DTYBAM	EF88
DUPLC1	85A5	DUPLCT	C8C1	DX0000	C90C	DX0005	C90F
DX0010	C923	DX0020	C928	DYFILE	=0040	E10	E718
E20	E722	E30	E727	E40	E735	E45	E739
E50	E73D	E55	E742	E60	E754	E70	E763
E90	E74D	EA10	E76D	EA20	E77E	EADV1	E767
EADV2	E775	ECHKSM	FEE6	END	F99C	END002	F9B1
END003	F9C1	END10	F9D9	END20	F9E4	END30	F9FA
END30X	FA02	END33	FABE	END33X	F9D6	END40	F9CB
ENDCMD	C194	ENDRD	D7EB	ENDSAV	C199	END_OF_FILE	914E
ENTFND	0253	ENTIND	00DD	ENTSEC	00D8	EOI	=0008
EOIFLG	00F8	EOIOUT	=0080	EOISND	=0008	EPTR	0299
ERO	FECD	ERO0	FECC	ER1	FECE	ER2	FECF
ER3	FED0	ERLED	026D	ERMOVE	E706	ERMSG2	E6E3
ERR	F553	ERR1	E625	ERR10	E698	ERR2	E627

* * Symbol Table * *

ERR3	E62D	ERR4	E644	ERRBUF	02D5	ERRCHN	=0005
ERRMSG	E6C7	ERROFF	C123	ERRON	C12C	ERROR	E60A
ERRR	F969	ERRR10	F975	ERRSA	=0010	ERRTAB	E4FC
ERRTSO	E6C1	ERWORD	026C	ESEEK	F410	ETEND	=E60A
EX	F36E	EX2	F370	EX3	F379	EXBAD	838C
EXCHN2	8594	EXCHNL	8591	EXDO	93A2	EXE	F367
EXEC	=00E0	EXECD	=0060	EXRD	83E8	FOOO	FB43
FOO1	FB46	FOO5	FB5C	FOO6	FB64	FOO7	FB67
FOO9	FB7D	FO13	FB97	FO14	FB9E	F1CNT	0277
F1PTR	00D3	F2CNT	0278	F2PTR	0279	FAIL	8381
FASTSR	0037	FB1	D2BC	FB2	D2C8	FB3	D2D8
FBAM10	FO04	FBG10	FE44	FBG15	FE64	FBTOG	FE30
FBUFS	=0300	FF10	C4AA	FF15	C492	FF25	C4BA
FF30	C4C9	FF40	C4D7	FFRE	C48B	FFST	C49D
FILCNT	0295	FILDRV	00E2	FILNOP	=0061	FILOPN	=0060
FILSEC	0285	FILTBL	027A	FILTRK	0280	FILTYP	00EC
FIL_SYN	9B73	FIN	F31D	FINAL	8387	FINBAD	8384
FINDBUF	91CA	FINMFM	8C35	FIXBAM	EFF1	FLOO	8CBA
FLO5	C6AD	FL10	C6C8	FLEXST	=0063	FLNTFD	=0062
FM2030	=0042	FM4040	=0041	FMHDPT	001B	FMT105	C8E0
FMT110	C8EF	FMTE10	FDDB	FMTEND	FD96	FMTERR	FDD3
FMTSID	01B2	FMTVAR	=0620	FND10	D383	FND2	F1F5
FND30	D391	FNDBUF	D2BA	FNDC20	D0F3	FNDC25	D0F9
FNDC30	D106	FNDFIL	C4B5	FNDL10	E1B7	FNDL20	E1C8
FNDL30	E1C4	FNDLMT	C6A6	FNDLNX	D37F	FNDLST	E1B2
FNDNO	F195	FNDN1	F19A	FNDN2	F19D	FNDNXT	F173
FNDRCH	D0EB	FNDREL	CE0E	FNDSEC	F1E6	FNDW10	D119
FNDW13	D10F	FNDW15	D121	FNDW20	D123	FNDWCH	D107
FNLOOK	89A1	FN_000	898A	FN_001	8996	FN_002	89B2
FN_INT	8986	FORMAT	C8C6	FORMT	FAC7	FOUND	028F
FRD_00	8394	FRD_01	83A3	FRD_02	83D2	FRE10	EF87
FRE20	EF7F	FRE25	D259	FREAD	=0088	FREB1	D2F9
FREB2	D303	FREBUF	D2F3	FRECHN	D227	FRECO	D22E
FREECO	C001	FREIAC	D2DA	FREICH	D4DA	FREMSG	C817
FRERD	D22E	FRERTS	EF87	FRETS	EF5F	FRETS2	EF62
FREUS2	EFD3	FREUS3	EFD5	FREUSE	EFCF	FREWRT	D22E
FRI10	D2E9	FRI20	D2D9	FRMERR	EA4E	FRMERX	E999
FRMT	=00F0	FS10	C3B0	FS15	C3B8	FS1SET	C398
FSTFMT	84B7	FSTLOAD	9080	FSTRD	8371	FSTSEK	848B
FSTWRT	83EC	FTNUM	0051	FWAIT	FB39	FWAIT2	FB3E
FWRT_0	8402	FWRT_1	8412	FWRT_2	8415	FWRT_3	8428
FWRT_4	8420	FWRT_5	8484	FWRT_6	8487	GA1	DF9B
GA2	DFA0	GA3	DFB0	GAFLGS	DF9E	GAP1	=000B
GAP2	=0004	GAPMFM	8C4F	GAPTBL	9ACA	GBERR	D20F
GBF1	D2A3	GBF2	D2B6	GBYTE	D39B	GCBYTE	CA39
GCRB10	F90C	GCRB20	F92B	GCRBIN	F8E0	GCRERR	0035
GCRFLG	0050	GCRHI	F8A0	GCRLO	F8C0	GCRPNT	0034
GCRTB1	A00D	GCRTB2	A10D	GCRTB3	A20D	GCRTB4	A30D
GCRTBA	9F0D	GCRTBD	9F0F	GCRTBE	9F1D	GCRTBF	9F0E
GCRTBG	9F2A	GCRTBH	9F00	GD1	ED7E	GE10	D433

* * Symbol Table * *

GE15	D43A	GE20	D443	GE30	D445	GET	D3AA
GETO	D3CE	GET00	D3B4	GET1	D3D3	GET2	D3D7
GET3	D403	GET4GB	F7E6	GET6	D409	GETACT	DF93
GETB1	D151	GETB2	D14D	GETBF	D1F8	GETBUF	D28E
GETBYT	D137	GETD3	ED6D	GETDIR	ED67	GETERC	D414
GETHDR	DE3E	GETINA	DFB7	GETLNK	DE0C	GETNAM	C6CE
GETPNT	D4E8	GETPRE	D12F	GETR2	D1E3	GETR3	D206
GETR4	D226	GETR5	D217	GETR52	D1F3	GETR55	D1F5
GETRCH	D1E2	GETSEC	F1FA	GETSIM	CD3C	GETWCH	D1DF
GI10	DFBF	GIB20	CA52	GIBYTE	CA35	GNO5	C6F7
GNO50	C6FC	GNO51	C70E	GN10	C71B	GN12	C728
GN14	C73C	GN15	C74A	GN20	C76B	GN22	C773
GN30	C783	GN35	C793	GN37	C798	GN40	C7A7
GN45	C7AB	GNSUB	C6DE	GOTU	F33C	GOTU10	F342
GO_FMT	8CDE	GP1	D4EB	GP2CMD	=0009	GP2TBL	9AD3
GS10	F20D	GS20	F21D	GS30	F21F	GSSPNT	DF45
GTAB	0056	GTABYE	E848	GTABYT	E839	GTSEC	8643
HANDSK	9228	HBID	0039	HDRPNT	0032	HDRS	0006
HEADER	0016	HEADS	8FB3	HED2TS	D552	HEDOFF	D676
HEXO	E69F	HEX5	E6AA	HEXDEC	E69B	HINIB	003B
HOF1	D67C	HOF2	D688	HOF3	D692	HSKRD	85F9
HSKSND	860B	HSKTST	85F6	IBOP	DF25	IBRD	DF1B
IBWT	DF21	ICMD	0096	ICR	400D	ID20	D00B
IDLO1	EC07	IDL1	EBFF	IDL10	EC81	IDL11	EC8B
IDL12	EC98	IDL2	EC12	IDL3	EC2B	IDL4	EC31
IDL5	EC3B	IDL6	EC58	IDL7	EC5C	IDL8	EC69
IDL9	EC6D	IDLE	EBE7	IER1	180E	IER2	1C0E
IFR1	180D	IFR2	1C0D	IGERR	83C9	ILERR	EA4E
IMAGE	028B	INAC10	FA0E	INAC20	FA1C	INACT	FA05
INCPNT	D1C6	INCPTR	D1C6	INDEX	0294	INFO	001B
INITDR	D042	INITP	DCB6	INNER	8B05	INQST	856B
INTDRV	D005	INTT1	EB4F	INTTAB	EB4B	INTTS	F1A9
IP	0075	IRQ	9D88	IRQJMP	02A9	IRQ_0	9DEA
IRQ_1	9DF2	IRQ_2	9E14	IRSA	=0011	ISRO1	E95C
ISRO2	E963	ISRO3	E980	ISRO4	E987	ISRCLK	E976
ISRHI	E973	IT20	D024	IT30	D02C	ITERR	EA4E
ITRIAL	D00E	ITS1	F1B8	ITS2	F1CB	ITS3	F1DF
IWSA	=0012	JACPTR	82C7	JAINSRV	80CE	JBMP	93B0
JCLEAR	9D63	JCNVBIN	952F	JDSTRT	9600	JEND	99CA
JERRR	99B5	JFMT	939F	JFMTE10	9D59	JFMTERR	9D51
JFORMAT	A40D	JFORMT	9B29	JFRMERR	8362	JGCRBIN	9965
JGET4GB	98D9	JILERR	8366	JIRQ	9DDE	JLCC	92BA
JLISTEN	8342	JMPC	=0050	JOB	0045	JOBN	003F
JOBNUM	00F9	JOBRTN	0298	JOBS	0000	JREAD	9606
JREAD01	960D	JSEAK	944F	JSETJB	93D1	JSETJB1	93D3
JSLOWD	A483	JSLOWER	A47E	JSRCH	970F	JSYNC	9754
JTALK	81EA	JTOP	92C8	JUMPC	=00D0	JVRFY	9898
JWRIGHT	976E	JWTOBIN	97F9	KILL	FE00	KILLP	F258
L213	FAF5	L214	FB00	L40	CFBF	L41	CFC9
L42	CFD8	L45	CFED	L46	CFCE	L460	F432

* * Symbol Table * *

L465	F461	L470	F483	L480	F43A	L490	F48D
L50	CFE8	LBUSED	0257	LCC	F2B0	LDO1	DA62
LDO2	DA6D	LDO3	DA86	LDO5	DA90	LD10	DA9E
LD20	DAA7	LDCMD	=000C	LEDO	=0008	LED1	=0000
LEDMSK	FECA	LEDOUT	=1C02	LEDPRT	=1C00	LEDS0	C110
LEDS1	C113	LEDSON	C118	LED_OFF	8788	LED_ON	877C
LIMIT	0276	LINDX	0082	LINTAB	022B	LINUSE	0256
LISNER	=0001	LISTEN	EA2E	LK05	C452	LK10	C45C
LK15	C462	LK20	C470	LK25	C475	LK26	C47E
LK30	C485	LOADIR	DA55	LOCK	02AF	LONGLN	=0032
LOOKUP	C44F	LRF	=0080	LRUEXT	CF1D	LRUILP	CEFC
LRUINT	CEFA	LRULP1	CF0D	LRUTBL	00FA	LRUUPD	CF09
LSN15	EA39	LSN30	EA44	LSNACT	0079	LSNADR	0077
LSNERR	E688	LSTBUF	0293	LSTCHR	0244	LSTDRV	028E
LSTJOB	025B	LSTSEC	024E	LT00	8CA2	LWPT	001E
LXINT	=000F	M10	CB50	M30	CB45	MAIN7	8B07
MAK10	FC3F	MAPOFF	FE87	MAPOUT	EEF4	MASK1	=00F8
MASK2	=0007	MASK2X	=00C0	MASK3	=003E	MASK4	=0001
MASK4X	=00F0	MASK5	=000F	MASK5X	=0080	MASK6	=007C
MASK7	=0003	MASK7X	=00E0	MASK8	=001F	MAX1	F24E
MAXIM	8978	MAXMIN	8961	MAXSA	=0012	MAXSEC	F24B
MAXSEK	0061	MAXTRK	02AC	MDIRTY	0251	MDMODE	=0003
MEM	CAF8	MEMERR	CB4B	MEMEX	CB1D	MEMRD	CB20
MEMWRT	CB50	MFMCMD	01B0	MFMER	8A82	MFMHDR	0024
MFMSIZ_HI	0044	MFMSIZ_LO	0271	MH10	D69A	MINIM	8966
MINSEK	0060	MINSTP	0064	MISTYP	=0064	MO10	EEFB
MODE	0297	MODES	9026	MODLST	FEB2	MOTOFF	8770
MOTON	8764	MOVB1	ED5B	MOVBUF	ED59	MOVHED	D693
MOVO10	FDF7	MOVOVR	FDF5	MOVUP	FDE5	MRCNF	8A4A
MRK1	EE04	MRK2	EDEE	MSCNT	=00FF	MSEEK1	8A37
MSEEK3	8A50	MSG1	C80B	MSGFRE	C806	MSGLEN	=000C
MTRCNT	02AB	MUL100	CE50	MUL200	CE57	MUL25	CE41
MUL400	CE6D	MUL50	CE4C	MULPLY	CE2C	MXCHNS	=0006
MXFILS	=0005	MYPA	0097	N101	EE19	N108	EE46
N110	EE56	NAMBUF	02B1	NB10	E2E9	NB20	E2F1
NB30	E303	NBCMDS	=0006	NBKH	00BB	NBKL	00B5
NBSIZ	=001B	NBTEMP	0272	NCMDS	=000C	ND10	C7DC
ND15	C7E5	ND20	C7ED	NDBH	02FC	NDBL	02FA
NDKRD	837F	NDKWRT	83F8	NEW	EE0D	NEWDIR	C7B7
NEWMAP	EEB7	NEWMPV	EEB7	NEWSS	E44E	NEXTS	004D
NFCALC	D075	NM10	EEC7	NM20	EED9	NM30	EEE3
NMI	FEE7	NMIFLG	0067	NMODES	=0004	NNMI	FF01
NNMI10	FF0D	NOBLK	=0065	NOCFIL	=0039	NOCHNL	=0070
NODRIV	=0074	NODRRD	Macro	NODRV	00FF	NODRWR	Macro
NOEOI	E94B	NOFILE	=0034	NOREC	=0050	NOTFND	=0006
NOTLK	E915	NOTRAN	83DD	NOTRDY	=0000	NOTRX	843B
NOTRX0	844D	NOTYET	D5C4	NOWRT	8479	NO_IND	8AF1
NO_MAX	8980	NO_MIN	896E	NO_TRK	891A	NR	00C1
NRBU20	E07B	NRBU50	E05D	NRBU70	E06B	NRBUF	E03C
NS00	8CA7	NS20	E4AC	NS40	E4D1	NS50	E4DE

* * Symbol Table * *

NSECTK	8A7E	NSSL	=0006	NSSP	=0078	NTYPES	=0005
NULBUF	E2E2	NULLNK	DE19	NUM	=0621	NUMF1	D07D
NUMF2	D083	NUMFRE	EF4D	NUMJOB	=0006	NUMSEC	FED1
NUMSEK	8C53	NUMSYN	=0005	NUM_SEC	942C	NXDB1	D4BB
NXDRBK	D48D	NXOUT	E009	NXT1	F12D	NXT2	F15F
NXTB1	D45F	NXTBF	004E	NXTBUF	D44D	NXTDS	F12D
NXTERR	F15A	NXTJOB	0041	NXTPNT	004F	NXTR15	DFE4
NXTR20	DFF6	NXTR30	E034	NXTR35	E035	NXTR40	E01D
NXTR45	E018	NXTR50	E02A	NXTREC	DFD0	NXTRK	0042
NXTST	0062	NXTTS	F11E	NZONES	FED6	OB05	CBA0
OB10	CBA5	OB15	CBB8	OB30	CBF1	OFFSET	FEDB
OK	D5C2	OKERR	E6BC	OKSEC	8641	ONEDRV	C312
ONEMS	8830	ONEMS1	8834	OPO2	D7CF	OPO21	D7F3
OPO4	D7FF	OPO41	D815	OPO415	D81C	OPO42	D82B
OPO49	D834	OPO5	D837	OP10	D83C	OP100	D94A
OP110	D95C	OP115	D965	OP120	D96A	OP125	D990
OP130	D9C3	OP20	D840	OP40	D876	OP45	D891
OP50	D8A7	OP60	D8B1	OP75	D8C6	OP77	D8CD
OP80	D8D9	OP81	D8E1	OP815	D8F0	OP82	D8F5
OP90	D940	OP95	D945	OPEN	D7B4	OPF1	DA06
OPFIN	D9EF	OPIR10	CA31	OPIRFL	C9FA	OPNBLK	CB84
OPNIRD	D475	OPNIWR	D486	OPNRCH	DC46	OPNTYP	D477
OPNWCH	DCDA	OPREAD	D9A0	OPTSCH	C3CA	OPWRIT	D9E3
OR10	DC65	OR20	DC81	OR30	DCA9	ORGSA	0084
OROW	DC98	OS10	C3D5	OS15	C3E8	OS30	C3EF
OS35	C400	OS40	C43C	OS45	C41B	OS50	C420
OS60	C434	OS70	C439	OVRBUF	=0100	OVRFLO	=0020
OW10	DCFD	OW20	DD16	P10	E2AA	P2	E289
P30	E291	P75	E2BF	P80	E2C2	PA1	1801
PARSE	C268	PARSXQ	C146	PATCH5	FF10	PATCH6	FF20
PATCH7	FF2F	PATCH9	FF45	PATFLG	028A	PATTYP	00E7
PB	1800	PBYTE	CFAF	PCMD	=0009	PCR1	180C
PCR2	1C0C	PD10	EA7E	PD11	EA8F	PD20	EA7F
PD21	EA90	PE20	EA74	PE30	EA75	PE40	EA8E
PEA7A	FEEA	PERR	EA71	PERR2	EB1F	PEZR0	EA6E
PHASE	02FE	PI1	DFCD	PIBYTE	CF9B	POSBUF	E29C
POSITN	E275	POTA1	180F	PPPPPP	FA38	PR10	C26B
PR20	C280	PR25	C283	PR28	C299	PR30	C29E
PR35	C2A0	PR40	C2B1	PRC00	86FD	PRC01	8706
PRC02	871B	PRC03	8722	PRC04	8729	PRC05	8730
PRC06	873A	PRC07	875F	PRC08	8760	PRCMD	86E6
PRGDRV	026E	PRGSEC	026F	PRGTRK	007E	PRGTYP	=0002
PRSCLN	C1E5	PS05	C160	PS10	C16A	PS20	C17A
PS30	C184	PTCH10	FF82	PTCH11	FF9C	PTCH12	FFA1
PTCH13	FFA6	PTCH15	FFAE	PTCH18	A5FE	PTCH19	A61C
PTCH20	A62B	PTCH21	A634	PTCH22	A65A	PTCH22R	FF18
PTCH23	A667	PTCH24	A6E5	PTCH25	A738	PTCH26	A74F
PTCH27	A764	PTCH28	A779	PTCH29	A786	PTCH30	A7B3
PTCH31	A7C0	PTCH32	A7C7	PTCH33	A83E	PTCH34	A874
PTCH35	A8A9	PTCH36	A8DB	PTCH37	A905	PTCH38	A91E

* * Symbol Table * *

PTCH39	A937	PTCH40	A951	PTCH41	A97F	PTCH42	A989
PTCH43	A99D	PTCH44	A9A5	PTCH45	A9AC	PTCH46	A9B9
PTCH47	A9D5	PTCH48	A9F8	PTCH49	AA02	PTCH50	AA17
PTCH51	AA1D	PTCH52	AA25	PTCH53	AA2D	PU10	EAAC
PU20	EAB2	PU30	EAB7	PUPS1	C932	PUT	CFB7
PUT4BG	F6D0	PUTB1	CFFD	PUTBAM	F0A5	PUTBYT	CFF1
PUTINA	DFC2	PUTSS	DD8D	QUE	F2F9	QUE05	F306
QUE10	F301	QUE20	F320	QUERDK	8517	QUIT	D635
QUIT2	D63F	R0	0086	R1	0087	R2	0088
R20	E219	R3	0089	R30	E228	R35	E253
R4	008A	R40	E253	R50	E265	R60	E272
RA10	EAF2	RA30	EB02	RA40	EB04	RAMEND	=0300
RAMTST	EAF0	RBM10	F0F2	RBM20	F10A	RD0	D15D
RD01	D164	RD05	E15E	RD1	D16A	RD10	E127
RD15	E138	RD20	E13B	RD25	E13D	RD3	D191
RD30	E14D	RD4	D192	RD40	E153	RDAB	DE57
RDBUF	D0C3	RDBYT	D156	RDGCR	83A0	RDIN	DE65
RDLNK	DE95	RDMAX	=0006	RDMODE	=0000	RDREL	E120
RDS5	DE75	RDSS	DE73	RDYLST	=0001	RDYTLK	=0088
REA7D	EA7D	READ	=0080	READ01	F4D1	READ11	F4D4
READ20	F4E1	READ28	F4FB	READ40	F505	REALUS	8065
REC	0258	RECO	D5F4	RECO1	D5E3	REC1	D600
REC3	D625	REC5	D631	REC7	D644	REC8	D651
REC9	D65C	REC95	D66D	RECH	00BB	RECL	00B5
RECORD	E207	RECOV	D5C6	RECOVF	=0051	RECPTR	00D4
REDBAM	F0DF	REED	F4CA	REL1	D26B	REL10	D253
REL15	D24D	REL2	D27C	REL3	D28D	RELBUF	D25A
RELINX	D249	RELP05	E096	RELP06	E094	RELP07	E09E
RELP10	E0A3	RELP20	E0AA	RELPTR	00D7	RELPUT	E07C
RELTYP	=0004	REMDR	=0627	RENAME	CA88	REQFST	8199
RESOFF	87AA	RESULT	008B	RES_PNT	A496	REVCNT	006A
RLINDX	0271	RM10	EAC9	RN1O	CA97	RNDEOI	=0081
RNDGET	D3DE	RNDRDY	=0089	RNGET1	D3EC	RNGET2	D3EE
RNGET3	D3FF	RNGET4	D3F0	ROM	=C000	RS	00C7
RSTEPS	0061	RT10	EAD6	RT20	EAD8	RTCH18	E7A8
RTCH19	EBFF	RTCH21	F9B1	RTCH25	F008	RTCH27	EE43
RTCH31	EB25	RTCH34	EF96	RTCH35	F1FD	RTCH36	F130
RTCH36A	F138	RTCH37	F1C7	RTCH37A	F1C9	RTCH38	F1E2
RTCH38A	F1E4	RTCH43	C1B7	RTCH44	C664	RTCH46	E60D
RTCH48	F266	RTCH51	D075	RTCH52	F01B	S00	8CB5
SA	0083	SA05	C370	SA10	C383	SA20	C388
SAVBM	02AD	SAVPNT	002E	SAVSP	0049	SAV_PNT	A48B
SB10	EF07	SB20	EF24	SBM10	F022	SBM30	F03E
SC15	C835	SC17	C855	SC20	C86B	SC25	C86D
SC30	C872	SCAL1	DF4C	SCFLG	DD95	SCHTBL	C440
SCR1	DDFC	SCRBAM	EEFF	SCREN1	C1AD	SCREND	C1A3
SCRTCH	C823	SCRUB	DDF1	SD20	C34C	SD22	C34D
SD24	C34F	SD40	C352	SD50	=C361	SDIRTY	E105
SDR	400C	SE10	E1D8	SE20	E1DC	SE30	E1EF
SEAK	F3B1	SEARCH	C617	SECINC	0069	SECKZZ	8A7A

* * Symbol Table * *

SECLES	88D1	SECLMO	88CB	SECSEK	=00B8	SECSS	025A
SECT	=0628	SECTCV	8888	SECTHI	8A76	SECTLO	8A72
SECTNX	886C	SECTOP	889E	SECTOR	0081	SECTR	0043
SECTSK	F6CA	SEEK	=00B0	SEEK10	F3BB	SEEK15	F3C8
SEEK20	F407	SEEK30	F3DC	SEKDN	87CC	SEKDUN	87D8
SEKE	87BA	SEKLOP	87C3	SEKRTN	87DD	SEKTR	861E
SEKUP	87D2	SEL_SID	8954	SEQGET	D400	SEQTYP	=0001
SETOO	DE2B	SETANY	C368	SETBAM	F011	SETBPT	EF3A
SETDIR	D4EB	SETDRN	D1D3	SETDRV	C33C	SETERR	EBD5
SETFLG	DD97	SETH	D6D3	SETHDR	D6D0	SETJ10	F3AB
SETJB	F393	SETJB1	F395	SETJOB	D50E	SETLO1	E17E
SETLO5	E19D	SETL10	E1A4	SETL40	E1AC	SETLDS	C100
SETLE	FA4E	SETLJB	D506	SETLNK	DDFD	SETLST	E16E
SETPNT	D4C8	SETSSP	DEE9	SET_BM	A4A1	SET_BUF	91DA
SET_FIL	91EA	SET_SIDE	93F3	SHORT	FA3B	SIDE	01B1
SIGN	8FB0	SIGNATURE	924E	SIGNATURE_HI	8001	SIGNATURE_LO	8000
SIG_ERR	92B3	SIMPRS	C1D1	SJ10	DE7F	SJ20	DE8B
SJ30	DE92	SJB1	D57A	SJB2	D535	SJB3	D538
SJB4	D53F	SKIP1	=0024	SKIP2	=002C	SLOWB	81AB
SLOWD	FEF3	SLOWE	FEF6	SP10	C1E2	SPDCHK	9ADC
SPINP	81B2	SPOUT	81CE	SR1	180A	SR10	C5C4
SR15	C5CA	SR2	1C0A	SR20	C5D7	SR30	C5FB
SR40	C629	SR50	C62F	SRCH	F510	SRCH20	F538
SRCH25	F53D	SRCH30	F54E	SRCHST	C5AC	SRRE	C604
SS	00CD	SSA10	FA94	SSACL	FA7B	SSCALC	DF51
SSDEC	FAA5	SSDIR	DEDC	SSEND	E1CB	SSIND	00D6
SSIOFF	=0010	SSNUM	00D5	SSP10	DF0B	SSP20	DF12
SSPOS	DEF8	SSRUN	FA97	SSSET	DED2	SSTEST	DF66
ST00	8CAD	ST10	DF77	ST20	DF7B	ST30	DF8B
ST40	DF8F	STAB	0024	START8	8CBF	STATQY	8581
STBCTL	9FB6	STBCTR	864B	STBCTT	865E	STBKNT	9FB8
STBRET	8683	STDIR	EC9E	STEPS	004A	STIN	87DF
STLO5	D33E	STL10	D348	STL20	D355	STL30	D35E
STL40	D363	STL50	D373	STL60	D37A	STLBUF	D339
STMTR	8794	STOUT	87E7	STP	FA69	STPFIN	8814
STPIN	FA63	STPOUT	FA32	STR1	D0B7	STRDBL	D0AF
STRRD	D09B	STRSIZ	024B	STRTIT	D0C9	STRTWD	884E
STRUCT	=FEA5	SWAP	F05B	SWAP1	F0BE	SWAP2	F0D0
SWAP3	F07F	SWAP4	F09F	SWITCH	003B	SYC10	FDB9
SYNC	F556	SYNC10	F55D	SYNCLR	FDA3	SYSIRQ	FE67
SYSTS	=0067	SYS_ERR	91AD	SZ00	8C95	S_LOG	892A
TO	=006F	TOV1	C3C7	T1	=0070	T1HC1	1805
T1HC2	1C05	T1HL1	1807	T1HL2	1C07	T1LC1	1804
T1LC2	1C04	T1LL1	1806	T1LL2	1C06	T2	=0071
T2HC1	1809	T2LC1	1808	T2LH2	1C09	T2LL2	1C08
T3	=0072	T4	=0073	TAGCMD	C1EE	TALK	E909
TALK1	E90F	TALK2	E925	TALKER	=0080	TATN20	EA6B
TBAM	029D	TC25	C1F3	TC30	C1F8	TC35	C200
TC40	C20A	TC50	C228	TC60	C245	TC70	C24C
TC75	C254	TC80	C260	TEMP	006F	TEMPSA	024C

* * Symbol Table * *

TGLBUF	CF8C	TIM	=0020	TIM2	=0040	TIMA_H	4005
TIMA_L	4004	TIMER1	=1805	TJ10	D5BA	TLERR	E68E
TLKO2	E937	TLKO3	E941	TLKO5	E916	TLKACT	007A
TLKADR	0078	TLKERR	E680	TMP	004B	TN10	C681
TN20	C687	TOBIG	=0004	TOFF	029A	TOGDRV	C38F
TOLONG	=0002	TOMANY	=0003	TOP	F2BE	TOPP	FB0C
TOPRD	=0045	TOPTST	9B89	TOPWRT	=0045	TOSMAL	=0005
TP1LST	FEC0	TP2LST	FEC5	TPLST	FEB6	TR10	C697
TR20	C6A5	TRACC	0040	TRACK	0080	TRAL	=0624
TRCMBF	C688	TRKNUM	FED7	TRKSS	0259	TRNAME	C66E
TRNOFF	F98F	TRYS	=0623	TSCHK	D55F	TSER1	D54D
TSERR	D54A	TSTO5	FD67	TSTOV1	C3BD	TST10	FD77
TSTA50	EA63	TSTATN	EA59	TSTC20	DDB9	TSTC30	DDC2
TSTC40	DDCA	TSTCHN	DDB7	TSTDAT	FD62	TSTFLG	DDA6
TSTJOB	D5A6	TSTRDJ	F47E	TSTRTN	EA62	TSTRTS	DDC9
TSTTRK	003B	TSTWRT	DDAB	TURNON	F97E	TYPE	024A
TYPFIL	D125	TYPFLG	0296	TYPLST	FEBB	TYPMSK	=0007
UBAM	029B	UBLKRD	CD5F	UBLKWT	CD97	UBLOCK	FFEA
UNLSN	=003F	UNTLK	=005F	UNUSED	806D	UPINST	85E9
US10	CB6C	USE10	EFBA	USE20	EFBD	USE30	EFB2
USEDTS	EF93	USER	CB5C	USERTS	EFCE	USREXC	CB72
USRINT	CB63	USRJMP	006B	USRTYP	=0003	UTLBAD	9021
UTLD00	E7C5	UTLD10	E7D8	UTLD20	E7FA	UTLD30	E802
UTLD35	E817	UTLD50	E82C	UTLODR	E7A3	V1	8B23
V2	8B7C	V3	8BB1	V4	8BEB	V5	8C4D
V6	8AF9	VAL	=0000	VALDAT	ED84	VD10	ED9C
VD15	EDB3	VD17	EDCB	VD20	EDD4	VD25	EDD9
VERDIR	ED84	VERERR	=0007	VERFMT	88F0	VERNUM	FED5
VER_00	8915	VER_FM	88F8	VFIN	8C48	VFMT	9CCF
VMKBAM	EDE5	VNERR	D572	VNMI	0065	VRF10	F698
VRF15	F6A5	VRF20	F6C5	VRF30	F6B3	VRFY	F691
WAITDN	8861	WATJOB	D599	WBAM	02F9	WDDAT	2003
WDSEC	2002	WDSTAT	2000	WDTEST	Macro	WDTRK	2001
WFREE	EF5C	WGP2	FD09	WHERE	A534	WLINDX	0270
WORK	0044	WORKTBL	9409	WPSW	001C	WR10	E0B2
WR20	E0B7	WR30	E0BC	WR40	E0C8	WR45	E0D6
WR50	E0D9	WR51	E0E1	WR60	E0E2	WRIGHT	F56E
WRITE	=0090	WRSTAT	8596	WRTO	D1A3	WRTO5	F575
WRT10	F586	WRT20	F58E	WRT30	F5B3	WRT40	F5BF
WRTAB	DE50	WRTBUF	D0C7	WRTBYT	D19D	WRTC1	D0E8
WRTGCR	840B	WRTMAX	=000C	WRTMIN	=0009	WRTN10	FDC9
WRTNUM	FDC3	WRTOUT	DE5E	WRTREL	E0AB	WRTS10	FCB8
WRTS20	FCC2	WRTS30	FCD1	WRTS40	FCEB	WRTS50	FCF9
WRTSNC	F5AB	WRTSS	DE6C	WRTSYN	FCB1	WRT_BAM	A58D
WSECT	F423	WTMODE	=0001	WTMTR	87B0	WTMTR1	87B2
WTOB14	F624	WTOB50	F643	WTOB52	F66E	WTOB53	F64F
WTOB55	F678	WTOB57	F685	WTOBIN	F5F2	WUSED	EF90
WVERFY	=00A0	XIDLE	836B	XMS	8829	XX05	F802
XX06	F85A	ZERRES	CED9	ZP2	=0065	ZPEND	=0103

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
A10	813A	776	784#										
A11	818D	787	840#										
A12	8177	818	827#										
A13	818A	828	836#										
A14	8171	821	824#										
A2	815A	738	810#	844	9022								
A3	80FA	743#	806										
A4	810D	746	755#										
A5	811E	756	766#										
A6	811B	753	764#										
A7	812C	767	775#										
A8	8155	773	782	805#									
A9	8192	764	797	843#	846								
ABRSK	8A65	3149	3198#										
AC10	F22B	18242#	18255										
AC20	F22D	18244#	18252										
AC30	F236	18248	18250#										
AC40	F246	18260	18265#										
ACC200	CEE6	11507	11554#										
ACCUM	008F	445#	11414	11421	11437$	11440$	11442$	11444	11446	11449$	11451$	11484	11487$
		11494	11495$	11500$	11503$	11505$	11513	11515	11516$	11518$	11520$	11521	11522
		11524	11533$	11554$	11555$	11556$	11568						
ACCX2	CEE5	11458	11509	11549	11553#								
ACCX4	CEE2	11512	11549#										
ACLSTP	0060	193#	20111$	20141$	20144$	20176$	20180$	20204$	20213$				
ACLTIM	0048	173#	2648$	5912$	6000$	6004$	8367$	19966$	20050$				
ACLTIM2	0035	142#	5925$	8369$	20063$								
ACPO0	E9DF	16636#	16643	21217									
ACPO0A	E9CD	16623#	16626										
ACPO0B	E9F2	16639	16647#										
ACPO1	EA0B	16645	16663#										
ACPO2	E9F7	16650#	16651										
ACPO2A	E9FD	16655#	16658										
ACPO3	EA0B	16664#	16668	16681									
ACPO3A	EA1A	16675#	16678										
ACPTR	E9C9	16387	16620#	16705									
ACR1	180B	327#	20305	20306$	20439	20440$							
ACR2	1C0B	18352#	18393$										
ACTJOB	001B	118#											
AD10	C325	9565#	9574										
ADD100	CEF0	11567#	11571										
ADDFIL	D6E4	10705	13085#	13490									
ADDNR	E304	14690	15284	15290	15302#								
ADDR1	E33B	10816	15347#										
ADDREL	E31C	14864	15332#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References					
ADDRES	CEED	11457	11508	11565#			
ADDRTS	DF65	14506	14508#				
ADDSUM	E84B	16242	16246	16259	16267	16345#	
ADDT12	DF5C	14492	14499	14502#			
ADRSED	007B	422#	5976	21272	21321$	21330$	
AF	005F	190#	18432$	20173	20201		
AF08	D715	13104	13112#				
AF10	D726	13113	13120#				
AF15	D730	13109	13124#				
AF20	D73D	13110	13115	13118	13123	13129#	
AF25	D74D	13134	13136#				
AF30	D76F	13152#	13155				
AF50	D790	13158	13168#				
AH10	CCAB	11166#	11182				
AH20	CCCA	11168	11170	11184#			
AH30	CCD0	11188#	11194				
AH35	CCD7	11193#	11197	11201			
AH40	CCE4	11190	11203#				
ALLDRS	C320	9563#	10351	10497	10822	13595	
ALLOC_B	A51E	8201#	8800				
AN05	E318	15306	15311	15316#			
AN10	E31B	15310	15319#				
AP30	DA42	13541	13544#				
APMODE	=0002	260#	13427	13446			
APPEND	DA2A	10689	13449	13531#	13535		
AR10	E355	15356	15360	15364#			
AR20	E363	15374#	15406	15409			
AR25	E368	15372	15377#				
AR30	E372	15381	15384#				
AR35	E38F	15397	15400	15403#			
AR40	E39D	15405	15410#				
AR45	E3B6	15423#	15468	15472			
AR50	E3C8	15420	15430#	15473			
AR55	E3D4	15429	15435#				
AR60	E3F9	15449	15455#				
AR65	E418	15464	15474#				
AR70	E444	15493	15495#				
AS	005E	187#	18429$	20106	20107	20110	20179
ASCHEX	CCA1	11149	11160#				
ATN	=0080	349#					
ATN100	E8ED	16458	16463#				
ATN110	E8FA	16464	16470#				
ATN120	E8FD	16424	16474#				
ATN122	E902	16394	16401	16435	16476#	16478	
ATN30	E884	16387#	16444				

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References								
ATN35	E891	16390	16396#							
ATN40	E89B	16397	16403#							
ATN45	E8A9	16404	16412#							
ATN50	E8B7	16413	16421#							
ATN95	E8D2	16410	16419	16443#						
ATNA	=0010	348#	734	814	16378	16454				
ATNIRQ	E853	6547	16354#	21051						
ATNLOW	EA56	16721#								
ATNMOD	007D	424#	728$	811$	16371$	16451$	16743			
ATNPND	007C	423#	719$	6593$	16356$	16362$	17098			
ATNS15	E87B	16381#	16384							
ATNS20	E8D7	9023	16382	16450#	16477					
ATNSRV	E85B	8648	16359#	16721						
AUTO1	C65F	10053	10056	10059	10062#					
AUTO2	C669	10042	10046	10074	10077#					
AUTOFG	0068	400#	10041							
AUTOI	C63D	1958	2029	4327	9721	9727	9734	9744	10036#	10997
AVCK	F220	8924	18211	18234#						
B02	DEB9	14344#	14348							
BOTOBO	DEA5	14332#	15524							
B2X10	F118	18012	18016#							
BA10	CD06	11225#								
BA15	CD19	11229	11236#							
BA20	CD1A	11238#	11247							
BA30	CD2C	11232	11248#							
BA40	CD31	11244	11251#							
BADCMD	=0031	2183	2287	4271	9292	10430	10933	11083	15654#	
BADFN	=0033	13359	15656#	17413						
BADID	F41B	18708	18711	18731#						
BADSYN	=0030	9437	10513	11086	13261	15653#				
BADTS	=0066	12782	15668#							
BAM	02A1	583#	17857	17859	17894$	17942	17945$			
BAM2A	F10F	11794	17638	17834	18009#	18020				
BAM2X	F119	8137	10295	10396	10471	18019#				
BAMSIZ	FE86	18240	21070#							
BAM_ONE	0146	495#	8240	8244						
BAM_PT	A4E7	8164#	8759	8794	8836					
BCD2	E6B4	15887	15889#							
BCDDEC	E6AB	15882#	15913							
BCJMP	CC63	11109	11111	11119#						
BCTAB	CC5D	11094	11116#	11117						
BD_INT	88BE	2862	2872#	2896						
BE05	CDAB	11328#								
BE10	CDBA	11332	11335#							
BFCNT	=0005	252#	253	253	534	11014	16957	16986	16988	17130

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
BFONLY	83D0	1963 1988#
BGTAB	F77F	19344 19354 19374 19384 19406 19421 19435 19457 19466#
BH10	E2DC	15252 15254#
BHERE	E2D0	15223 15246#
BHERE2	E2D3	15231 15248#
BID	0038	148# 5509$ 5790$ 18910 19150$ 19823$
BIGFIL	=0052	15375 15661#
BINGO7	F7BA	19545# 19568
BING20	F7D9	19554 19571#
BING35	F76F	19448 19454#
BINGCR	F78F	5407 5634 6256 19037 19284 19512# 20620
BITCNT	0037	147#
BKOTST	CDF2	11257 11291 11318 11370#
BLINDX	=0006	258# 8215 11594 16991$ 16992$ 18010
BLK10	CC26	11083# 11098
BLK30	CC2B	11086# 11090 11155
BLK40	CC30	11081 11089#
BLK50	CC38	11094# 11097
BLK60	CC42	11095 11100#
BLKALC	CD03	11119 11222#
BLKEXC	CDA3	11123 11323#
BLKFRE	CCF5	11120 11214#
BLKNB	C7AC	10225 10285# 10297 10335
BLKNB1	C7B0	10287# 10289
BLKPAR	CC6F	11104 11126# 11282 11317
BLKPTR	CDBD	11124 11338#
BLKRD	CD56	11121 11275#
BLKRD2	CD36	11257# 11265 11325
BLKRD3	CD42	11265# 11276 11283
BLKTST	CDF5	11214 11223 11374#
BLKWT	CD73	11122 11291#
BLOCK	CC1B	11077# 21082 21090
BMASK	EFE9	8179 8250 8762 8797 12297 12306 17677 17720 17768 17772# 18247
BMINDX	A5DB	8173 8242 8319#
BMP	F37C	18589 18606#
BMPNT	006D	415# 8110 8112 8119$ 8121$ 8154 8194 8196$ 8208 8210$ 8220$ 8222$
		8236 8241$ 8245$ 8249 8300$ 8429$ 8432$ 8449 8474 8671$ 8726$ 8731$
		8738$ 8956 11884 17500$ 17511$ 17533$ 17536$ 17550$ 17562$ 17611$ 17643$ 17645$
		17676 17678$ 17682 17684$ 17719 17721$ 17724 17727$ 17767 17808$ 17858$ 17861$
		18053 18155 18180 18246 18258
BOOT	E780	16157#
BOOT2	E77F	16153#
BOOT3	E780	16170#
BOOT4	E7A8	16207#
BPO5	CC7C	11012 11130 11133# 11145

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
BP10	CC8B	11135	11138	11143#	11153								
BP20	CC92	11141	11149#										
BREAK	E202	15085	15102	15111#	15200								
BT05	CE08	11382#											
BT15	CDE0	11359#	11364										
BT20	CDE5	11357	11362#										
BTAB	0052	183#	5125	5128	5131	5137	5139	5179$	5182$	5188$	5190	5193	5205$
		5208$	5214$	5216	5219	5224	5227$	5230$	5236$	5238	5241	5244	5246$
		5254	5257$	5261	5272$	5277$	5278	5281	5288	5291$	5294$	5301	5304
		5305$	5313	5508	5513	5517	5521	5533	5537	5542	5546	5553	5557
		5567	5571	5575	5579	5750$	5756$	5762$	5768$	5789	5793	5797	5801
		5811	5816	5820	5824	18841	18844	18847	18853	18855	19149	19154	19158
		19162	19172	19176	19181	19185	19194	19198	19208	19212	19216	19220	19337
		19351	19367	19381	19399	19418	19428	19454	19529$	19534$	19538$	19542$	19552$
		19557$	19561$	19565$	19572$	19575$	19576$	19715$	19721$	19727$	19733$	19822	19826
		19830	19834	19844	19849	19853	19857	19864	19898$	19901$	19904$	19907$	19912$
		19915$	19918$	19919$	21006$	21010$	21014$	21018$					
BUF0	00A7	463#	8216	10397	10473$	11046$	11064	11713	11715$	11796	11812$	12160$	12170$
		12219	12224$	12320	12327	12335$	12408	12418	12447$	13910	14566	14585	14616
		14638	14640$	14905$	14920$	15572$	16980$	16987$	16989$	16991$	17968	17985$	
BUF1	00AE	464#	11716	11718$	12161$	12175$	12228	12233$	13918	13920$	14568	14592	14618
		14642$	16981$	16992$									
BUFF0	=0300	211#	20534$	20540$	20544$	20548$	20552$	20556$	20558$	20562	20563	20564	20565
		20567$	20594	20653	20768	20924	20925$	20929	20930$				
BUFF1	=0400	212#											
BUFF2	=0500	213#	20587$	20614	20825								
BUFIND	FEE0	4531	8219	8431	10305	11330	12711	13768	14337	14339	14359	14393	16953
		17642	21145#										
BUFPNT	0030	137#	3791$	3793$	3815$	3826$	3848$	3850$	3852	3866$	3881$	3883$	3903$
		3914$	3929$	3931$	3956	3968$	4012$	4014$	4036	4049$	4072$	4074$	4103$
		4875$	4877$	4878	4915$	4918$	5089$	5110	5112	5116$	5118$	5143$	5145$
		5220$	5242$	5262$	5282$	5456	5492$	5495	5499$	5554$	5558$	5568$	5572$
		5576$	5580$	5590	5595	5601	5609$	5650	5676	5685$	5688	5701	5713
		5731$	5733$	5735	5784	5831$	6215$	6242$	6245$	6337	6434	18600$	18602$
		18603	18638$	18642$	18801$	18826	18828	18832$	18834$	18859$	18861$	18894$	19082
		19116	19133$	19136	19140$	19195$	19199$	19209$	19213$	19217$	19221$	19231	19236
		19242	19250$	19300	19363$	19395$	19411$	19446$	19451$	19459$	19513$	19522	19526$
		19607	19614	19623$	19626	19634	19639	19649	19658	19665	19674	19680	19691$
		19694	19703	19817	19867$	19888	19892$	19924$	20595$	20615$	20707	20766$	20769$
		20783	20792	20794$	20937$	20990$	20997	21001$					
BUFS	=0300	210#	211	212	213	404#	405	4874	4914	5088	6156	6240	10438$
		10441$	10443$	18599	18637	18800							
BUFTAB	0099	461#	462	9491	9530$	10121	10123	11069$	11261	11343$	11779$	11780$	11857$
		11858	11951$	12037	12039	12043$	12045$	12047	12048$	12119$	12279$	12524	12529$
		12531$	12533	12536	12581$	12600$	12601	12603$	12677	12680$	12700	12702	13609$

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		13916$	13926$	14230	14411$	14413$	14684	14707	14789$	14791	14811$	14942	14946$
		14953	14958	15811$	16951$	16954$	16961$	16964$	16967$	16970$	17240$	17241$	17261$
		17277$	17278$	17286	17473$	17887$	17893	17896$	17897$	17936$	17943$	17946$	
BUFTST	CDD2	11338	11350#	11370									
BUFUSE	024F	515#	10469$	11029	11032	11036	11037$	11039	11040$	12296	12305	12307$	12342$
		12343$	17006$	17008$									
BUMP	=00C0	92#	4859	12965									
BUMPC	=0040	208#	18588										
BURST	86A0	2080	2451#	3987									
BURST_ROUTINES	8030	620#	10954										
BURTST	869D	2450#	2457										
BW10	CD81	11297	11300#										
BW20	CD8C	11307#											
BYTCNT	0036	144#	5504$	5511	5525$	5531	5550$	5561$	5565	5583$	5776$	5792	5805$
		5809	19145$	19152	19166$	19170	19189$	19202$	19206	19224$	19515$	19531	19545$
		19549	19809$	19825	19838$	19842	20992$	21003	21022$				
BYTE	003C	157#											
BYT_CLR	=1C00	231#	5416	5436	6140	6275	6288	6304	6318	6329	6339	6351	6372
		6377	6502										
CB	=00A3	462#	9284	9319$	9456	9622	9658	10925$	10927$	12569$	15892$	15910$	15912$
		15915$	15923$	15928$	15937$	16009$	16015$						
CB10	DECC	14364#	14367										
CBMV2	=0073	12810	15673#										
CBPTR	=000A	253#	461	462	9491	9530$	13609$	15811$					
CC10	C594	9844	9942#	9950									
CC15	C59A	9943	9946#										
CC20	C5A6	9948	9951#										
CCHKSM	C000	9211#											
CDIRTY	E115	14299	14917#										
CDRIVE	003E	159#	2642$	4749	4752$	4791	5049	5902	5931$	18411$	18485	18490$	18695
		18703	18762	19972	20040	20074$							
CHANNEL	02B0	590#	4386	4424	4546$								
CHAR	0275	548#	9285$	9288	9453$	9458							
CHCSEE	891B	2939#	3875	4002									
CHGUTL	8FE5	711	4228#										
CHKB10	F5EC	19116#	19118										
CHKBLK	F5E9	5312	5397	5631	6254	9172	18917	19027	19113#	19281	20618		
CHKIN	CACC	10670	10855#	10872									
CHKIO	CAE7	10699	10831	10871#									
CHKSEE	85AF	2002	2099	2293#									
CHKSUM	003A	150#	5252$	5258	5266$	5274	5398$	5632$	6196$	6199	6200$	6255$	18919
		19028$	19282$	19571	19865$	20619$							
CHK_BLK	A545	8227#	8828	8934									
CHNDAT	023E	507#	953	981	983$	11065$	11285$	12484	12535$	12547$	12587$	13603$	13898$
		14955$	14975$	15918$	16545	16547$							

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
CHNRDY	00F2	483#	898	919	1103	10780	11062$	11273$	12063$	12198$	12499	12508$	12512$
		12541$	12585$	13548$	13900$	13958$	14957$	14964$	14977$	15151$	15939$	16491	16515
		16694	17002$	17004$	17265$	17312$							
CJUMPH	FEA1	9302	21089#	21095									
CJUMPL	FE95	9197	9300	21081#	21087								
CK10	CAD6	10861#	10865										
CK20	CAE6	10863	10868#										
CK25	CAEA	10873#	10878										
CK30	CAF4	10874	10877#										
CKM1	DA11	13512#	13517										
CKM2	DA1C	13514	13519#										
CKT1	DA1E	13521#	13526										
CKT2	DA29	13523	13528#										
CKTM	DA09	13283	13291	13508#									
CLB1	F00B	17807#	17810										
CLD2	D334	12364	12372	12374#									
CLDCHN	D313	11833	12359#	17155	17169								
CLEAR	FE0E	20634	20956#										
CLER10	FE26	20969#	20972	20975									
CLKDAT	FEFB	1002	16562	21175#									
CLKHI	E9B7	730	909	996	2054	16373	16504	16556	16602#				
CLKIN	=0004	346#	740	850	1029	1046	1060	1088	16383	16625	16642	16657	16677
CLKLOW	E9AE	831	933	16467	16529	16595#	21175						
CLKOUT	=0008	347#	2037	3889	16596	16603							
CLNBAM	F0D1	9179	10470	11832	17530	17955#							
CLOSE	DAC0	800	4485	4499	4510	9070	9190	13616#	16438				
CLR10	E104	14881	14889#										
CLRB2	C1C1	9343#	9346										
CLRBAM	F005	8291	8589	9178	17508	17797#							
CLRBUF	DEC1	13977	14357#	15517									
CLRC1	D30B	12354#	12356										
CLRCB	C1BD	9337	9340#	15805									
CLRCHN	D307	2151	3627	12352#	17433								
CLREC	E0F3	14846	14879#	14887									
CLRF10	DDA3	14062	14067#										
CLRFLG	DD9D	11730	14058	14064#	14658	14688	14868	15135					
CLS05	DAD1	13624#	13632										
CLS10	DAD4	13620	13626#										
CLS15	DAE9	13635	13637#										
CLS20	DAF0	13643#	13646										
CLS25	DAFF	13648	13650#										
CLSALL	DAEC	13628	13640#										
CLSC28	DB0C	13657	13659#										
CLSC30	DB26	13665	13674#										
CLSC31	DB29	13670	13676#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
CLSCHN	DB02	10741	13629	13644	13653#								
CLSD	D317	12361#	12375										
CLSD4	DC06	13791	13797#										
CLSD5	DC21	13774	13813#										
CLSD6	DC29	13777	13812	13818#									
CLSDIR	DBA5	13673	13702	13750#									
CLSR1	DB5F	13701	13703#										
CLSREL	DB2C	13667	13679#										
CLSW10	DB76	13711	13715	13719#									
CLSW15	DB88	13728	13730#										
CLSW20	DB8C	13722	13734#										
CLSWRT	DB62	13672	13707#										
CMD	024D	512#	1923$	2007$	2120$	2442$	11942$	12617$	12729	12742$	12772	12816$	12835$
		12836	14268$	14272$	14276$	14280$	14284$	14288$	14466$	18100$			
CMD10	8E66	3948#	3954	3962	3969								
CMD100	8E7F	3959	3964#										
CMD7	8AF3	3317#	3321	3326									
CMD70	8B09	3336#	3340	3345									
CMD71	8B1D	3352#	3356	3361									
CMD72	8B97	3455#	3459	3465									
CMD73	8BAB	3473#	3477	3484									
CMD74	8BBF	3491#	3495	3501									
CMD750	8BE5	3524#	3528	3538	3542								
CMD75X	8BFE	3535	3540#										
CMD7A	8BD1	3507#	3511										
CMD7B	8B86	3440#	3444										
CMD7C	8B76	3427#	3431										
CMD7D	8B66	3414#	3418										
CMD7E	8B4E	3395#	3399										
CMD7F	8B3E	3382#	3386										
CMD7FE	8C1A	3567#	3571	3576									
CMD7FF	8C02	3549#	3553										
CMD7N	8B2F	3368#	3372										
CMD9EX	8DF3	3870	3875#										
CMDBUF	0200	504#	626	648$	1965	1967	1995$	2067	2069	2090$	2104	2130	2134
		2146	2148	2186	2272	2304	2369	2380	2390$	2440	2515	2520	2640
		2824	2835$	2844	2846$	2849	2851$	2853	2854$	2855$	2859	2864	2869
		2872$	2874$	2876$	2883	2886	2899$	2901$	2902$	2917	2926	2943	2993
		2996	3005	3008	3019	3034	3420	3433	3531	3558	3578	3640	3662$
		3668$	3672$	3675$	3678$	3682$	3685$	3711	3715	3721	3750	3764	3765
		3795	3797	3869$	3933	3935	3996$	4016	4018	4093	4129	4147$	4184
		4188	4201	4232	4275	4336	4393$	4563	4567	4576	4584	4585$	9090
		9094	9098	9344$	9587	9589	9600	9862	9878	10127	10145	10502	10886
		10890	10892	10896	10916	10937	10940	10950	11093	11133	11166	11757	13201
		13328	13351	13510	13567	13579$	13580$	15126	15145	15147	15153	16960	16963

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		17428	17430	21180									
CMDCHN	=0004	256#	11591	11612	11754	16987$	16996	17002$					
CMDCTL	806E	647	661#										
CMDEIG	8C57	2493	3618#										
CMDELE	8EC6	2496	3975	4011#									
CMDER2	E645	8265	9073	11250	12783	12811	13796	15113	15802#	16289	16326	18170	18195
		18267											
CMDER3	E648	8135	11806	15799	15804#	17832							
CMDERR	C1C8	2184	2288	4272	4518	9293	9312	9352#	9388	9438	9525	9731	10431
		10514	10681	10867	10876	10934	11009	11084	11087	11253	11360	11687	11775
		12169	12467	13262	13357	13360	13407	13416	13422	13469	14850	14980	15132
		15140	15211	15376	15499	16221	17414	17978	18076				
CMDFIV	8A27	2953	2959	3139	3151#	4136	4144						
CMDFOR	8A08	2489	3121#										
CMDFVE	8A09	2490	3135#										
CMDINF	86B9	2482#	2506										
CMDIS	=001E	2501#											
CMDJMP	86C8	2485#	2501	2562	2564	9202							
CMDLEN	=0029	278#	504	9341	9520	11757							
CMDNIN	8D67	2494	3786#	3873									
CMDNUM	022A	505#	9294$	9299	9432	9523$	11103$	11106	13200$	13559$	15791	15795$	
CMDONE	89EF	2486	2958	3102#	3135	3702	3710	4124					
CMDRET	8FAA	4188#	4237										
CMDRST	C2DC	9183	9522	9527#	13587								
CMDSA	=000F	259#	11975	16997$									
CMDSEC	8FA4	4184#	4234										
CMDSET	C2B3	9182	9283	9491#	13199	15125							
CMDSEV	8A86	2492	3225#	3735									
CMDSIZ	0274	547#	622	2176	2293	2939	3628	4228	4558	4561$	4580$	4588	9454
		9479	9518$	9626	9628	10155	10912	11000	11144	11181	13561	17426	
CMDTBB	808E	632	634	664#	9187								
CMDTBL	FE89	9287	21075#	21079									
CMDTEN	8DF6	2495	3880#	4000									
CMDTHI	8F5F	2498	4122#										
CMDTHR	8A03	2488	3116#										
CMDTWO	89FD	2487	3111#										
CMDTWV	8F18	2497	2919	4071#									
CMDWAT	0255	520#	11765$	17078	17082$								
CMDZER	89EC	2485	3099#										
CMD_11	8EDC	4020#											
CMD_12	8F25	4077#											
CMD_SEC	0096	448#	4142$	4152									
CMD_TRK	0067	398#	2222	2307$	2317	2319$	2325	2327$	2516$	2667	2672	2944$	2964
		3107$	3116$	3726	3727$	3753$	3769$	4130$	4156$				
CMPCHK	C589	9840	9937#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References										
CMPR10	FD39	20774#	20834									
CMPR15	FD40	20779#	20789									
CMPR20	FD58	20785	20798#	20814	20826							
CNT	=0620	6115$	6474$	20227#	20265$	20907$						
CNT10	FBBB	20417#	20419	20423	20425							
CNT20	FBCE	20418	20430#									
CNTINT	F259	9194	18372#	21295								
CNTST	=F259	18301#	18361									
CNVBIN	F497	18683	18826#									
CODE	FAC7	20244#										
COMP	FD2C	20765#	20799									
COMPAR	C4D8	9776	9816	9836#								
CONHDR	F934	5343	9167	18961	19888#							
CONSEK	8A57	3191#	3641	3669								
CONT	0098	453#	972$	1004$	1025$	1091$	16536$	16564$	16621$	16680$		
CONT10	F2C0	18463#	18503									
CONT20	F2F3	18465	18502#									
CONT30	F2CD	18468	18474#									
CONT35	F2D8	18476	18482#									
CONT40	F2E9	18486	18494#									
CONT50	F2F0	18495	18500#									
COP01	C982	10676	10679#									
COP05	C987	10678	10682#									
COP10	C9A1	10656	10660	10664	10668	10693#						
COPY	C952	10518	10652#									
COUNT	FBB6	20396	20413#									
CP00	8C90	3632	3661#									
CP02	C4E6	9842#	9845									
CP05	C4E7	9844#	9851	9854	9867	9870	9888					
CP10	C4EC	9841	9846#									
CP20	C4FE	9849	9856#									
CP30	C50A	9861#	9881									
CP32	C51B	9864	9871#									
CP33	C51D	9860	9874#									
CP34	C52B	9876	9882#									
CP40	C535	9880	9884	9890#								
CP42	C55C	9907	9910#									
CP50	C589	9929	9936#									
CPMINT	84F1	676	677	2167#								
CPMIT	003C	156#	2172	2187$	2381	2826	21298$					
CPMSEK	0097	451#	2218	2889$	2990	3002	3017	3024	3040	3141$	4134$	4146	4148$
CR	=000D	251#	9503	9514	12564	13717	14709	14974				
CR20	EAEA	16877#										
CR30	EAEC	16878#										
CRA	400E	386#	864	866$	881	883$	948	950$	958	960$	8626$	8627$	8634$

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
CRTDAT	FC88	20587#	20589										
CSO7	C2CA	9494	9517#										
CSO8	C2CB	9492	9504	9515	9518#								
CS10	C2FD	9541#	9549										
CSECT	004C	177#	5028$	5070	5084$	18740$	18783	18797$					
CSERR	F41E	18693	18734#										
CTL_CMD	005F	189#	1924$	1970	2392$	2441$	3027$	4394$	4398	4432$			
CTL_DAT	0046	170#	1990$	2015$	2064	2173$	2211$	2219$	2223$	2227$	2231$	2235$	2262$
		2333$	2336	2338$	2356	2378$	2382	2384	2389	2533$	2823$	2834	3031$
		3035	3622	3853$	3925	4452$	4472$	4480$	4495$	4501	4505$	4511	
CTRACK	0046	171#											
CTR_ERR	9199	4404	4494#	9045									
CURBLK	DE3B	9995	10031	12644	13095	13396	13439	14247#					
CUR_TRK	0064	199#	2301$	2669	2674	2693	2725$	2728	2732$	2966	3103$	3117$	3188$
CVSTAT	883C	2767#	3184	3831	3983								
CY	C9A7	10694	10698#										
CY10	C9B9	10691	10708#	10738									
CY10A	C9CE	10715	10718#										
CY15	C9D5	10721#	10727										
CY20	C9D8	10720	10723#										
CY30	C9EA	10730	10733#										
CYEXT	CA53	10716	10792#										
C_100	8E34	3886	3912	3917#									
C_101	8E04	3888#	3908	3915									
C_102	8E17	3894	3898#										
C_103	8E0F	3893#	3900										
C_104	8E2B	3905	3910#										
C_105	8EB7	3919	3991	3996#									
C_106	8EC2	3997	4002#										
C_107	8EC5	3994	4004#										
C_108	8E88	3966	3971#										
C_109	8E9A	3976	3982#										
C_110	8E93	3939	3951	3973	3978#								
C_1100	8EE1	4027#	4033	4043	4050								
C_1101	8F05	4030	4037	4052#									
C_1102	8EFC	4040	4045#										
C_1103	8EFE	4047#											
C_1104	8F10	4047	4063#										
C_111	8EA0	3980	3984#										
C_112	8E45	3923	3928#										
C_1200	8F2A	4084#	4090	4097	4104								
C_1201	8F4C	4087	4094	4106#									
C_1202	8F41	4099#											
C_1203	8F42	4100#											
C_1204	8F57	4100	4117#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
C_800	8D0D	3723	3730#	3758									
C_801	8D5F	3706	3719	3733	3736	3740	3743	3747	3777#				
C_802	8D40	3751	3760#										
C_803	8D5C	3761	3767	3775#									
C_900	8D89	3806#	3812	3820	3827								
C_901	8DA2	3817	3822#										
C_902	8DAB	3809	3824	3829#									
C_904	8DC6	3788	3847#										
C_905	8DBF	3835	3838	3842#									
C_906	8DE8	3845	3864	3869#									
C_907	8DD0	3852#	3860	3867									
C_908	8DDF	3857	3862#										
DATA	0085	432#	1076$	1084$	1095	10778$	10911$	11745	11750	11759	12058$	12064	12080
		12085	12565$	12575$	12586	14686$	14710$	14788	14833	14862	14866$	14884$	14969
		16284	16328	16673$	16684	17266	17307$	17313					
DATA2	1C01	4961	4971	5176	5187	5202	5213	5235	5251	5271	5299	5353	5383
		5431$	5449$	5460$	5643	5655	6139$	6274$	6287$	6303$	6317$	6328$	6338$
		6350$	6401	6428	6438	6503$	18335#	18669	18676	18893	18903	18972	19003
		19059$	19076$	19086$	19293	19305	20296$	20637$	20654$	20669$	20685$	20696$	20708$
		20719$	20782	20811	20824	20873$	20965$						
DATHI	E99C	830	990	1032	1056	16466	16554	16581#	16628	16653	21176		
DATIN	=0001	344#	907	915	925	930	936	977	1009	16502	16510	16521	16526
		16532	16540	16569									
DATLOW	E9A5	731	987	1050	1094	16374	16551	16588#	16647	16683			
DATOUT	=0002	345#	16582	16589									
DBID	0047	172#	5308	18418$	18911	19528							
DBL05	CF57	11656	11659	11664#									
DBL08	CF5D	11662	11667#										
DBL10	CF66	11653	11673#										
DBL15	CF6C	11637	11678#										
DBL20	CF6F	11672	11679#										
DBL30	CF76	11640	11685#	11696									
DBLBUF	CF1E	9188	11634#	11929	11932	12066	12075	12079	12105	13681	13724	13746	14745
		14755	14768	14770	15013	15019	15021	15233	15424	15431	15491	15513	15605
DBS10	CF8B	11694	11698#										
DBSET	CF7B	11691#	15336										
DBSYNC	FCE0	20682#	20688										
DDRA1	1803	319#	21195$										
DDRA2	1C03	5422$	5473$	6495$	18339#	19050$	19101$	20871$	20950$	20962$			
DDRB1	1802	318#	17060$	21202$									
DDRB2	1C02	18338#	18373$										
DEALL_B	A508	8187#	8765										
DEBNC	E9C0	849	906	914	924	929	935	975	1008	1028	1045	1059	1087
		16498	16509	16520	16525	16531	16538	16568	16609#	16611	16624	16641	16656
		16676											

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
DECTAB	CCF2	11196	11211#										
DEL1	C8AD	10411	10414#										
DEL10	C894	10399	10403#										
DEL2	C894	10404#	10416										
DELDIR	C8B6	10364	10418#	13345	17381								
DELFIL	C87D	10372	10382	10394#	13811								
DELIND	0292	569#	9762$	9804$	10000	10001$	10006	13107$	13121$	13128$	13129	13177	17343$
DELSEC	0291	568#	9966$	9993	9997$	10556$	13112	13126$	13174				
DERR	F1F5	18122	18193#										
DIAGOK	EB22	16913	16917#	17026	17028								
DIND	0266	536#	13179$	13395$	13482$	13756	14127						
DIR1	ECEA	17244#	17256	17317									
DIR10	ED0D	17257#	17290										
DIR3	ED23	17248	17269#										
DIRBUF	0094	446#	4408	4418	4425	4435	4449	4457	4462	4469	4476	4532$	4541
		4548$	9863	9868	9886	9900	9923	9926	9931	9990	9999	10029	10106$
		10122$	10124$	10128$	10209	10214	10227	10254	10306$	10308$	10311	10317	10360
		10366	10370	10420$	12120$	12560	12678$	12681$	12701$	12703$	13152$	13161$	13164$
		13167$	13380	13382$	13386$	13389$	13432	13434$	13455	13458	13461	14167$	14170$
		14186	14189	14207$	14214$	14231$	14233$	14379	14389$	14394$	14468	14471	14529
		14543$	14544	15049	15079$	15087	15100$	15101	15104	15219	15221	15280$	15289$
		15481$	15486$	15529	15536$	15546$	15550$	15553$	15556$	15578$	15581$	15594$	15597$
		17357	17360	17363	17367	17379	17486$	17489$	17494$	17497$			
DIRERR	=0071	8264	15671#	18194	18266								
DIRLEN	=0018	276#	10207	10212	10216								
DIRLST	0254	519#	12551	13622$	17264$								
DIRSEC	0290	567#	10012	10033$									
DIRTRK	FE85	8310	8459	8540	9970	10010	10748	11819	13233	13761	17348	17501	17686
		17729	17991	18057	18067	18080	18137	18159	21069#				
DIRTYP	=0007	265#	11070	11070	12504	13664							
DISKIN	89B3	2499	3052#	3136	3705								
DIV100	CE77	11484#	11491										
DIV120	CE71	11416	11481#										
DIV150	CE87	11493#	11523										
DIV200	CE89	11494#	11498										
DIV254	CE6E	11413	11479#										
DIV300	CEA3	11502	11508#										
DIV400	CEB0	11511	11513#										
DIV500	CEBF	11517	11519	11521#									
DIV600	CED6	11529	11531	11533#									
DIV700	CED8	11527	11534#										
DKIT10	EAA7	16809#	21203										
DKMODE	005E	186#	1949	2020	2118$	2144$	2193	2210	2254	2257$	2261	2273$	2334
		2337$	2508	2510$	3198	3202$							
DOIT	D590	12835#	13829										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
DOIT2	D593	12621 12836# 14476
DOITT	F473	18797# 18807
DOJOB	D58C	8489 8514 12828 12831#
DONE	F418	18728# 19313 19320
DOREAD	D586	8312 8420 8440 8461 12826#
DOREC	D6A6	12892 12927 12979 12989 13036#
DOREC1	D6AB	13040# 13056
DOREC2	D6B9	13046#
DOREC3	D6C4	13053 13057#
DOSTEP	FA2E	20038 20103 20117# 20134 20187
DOSVER	=0002	272# 17493
DOWRIT	D58A	8279 8288 8305 12829#
DO_THE_COMMAND	8761	2567 2587#
DRDBYT	D4F6	9984 11647 11650 12708# 14696 15412
DRIVE	003D	158# 18483$ 18489 18538 18585 18934 20248 20531
DRT	D466	12611 12616#
DRTRD	D460	11258 12610# 13117 13766
DRTWRT	D464	11307 11319 12615# 13168 13397 13440 17512
DRVCNT	028C	563# 9710$ 9718 9726$ 9741$ 9765$ 9810$ 9853 10549$
DRVFLG	028D	564# 9685$ 9770$ 9799$ 10183 10189 10192$ 10194$ 10202$
DRVMSK	=0001	279# 4741 4790 5048
DRVNUM	007F	426# 8126 8770 8805 9108 9128 9225 9275$ 9323 9637$ 9642 9645$
		9716$ 9846 9920 10044 10063 10078 10300 10447 10554$ 10703$ 10747$ 10834$
		10994$ 11048 11377$ 11835 11859 11898 12130$ 12272 12371 12616 12832 12882
		12886$ 12904 12952 12966 12987 12994$ 13024 13076 13100$ 13182 13212$ 13221
		13238 13249$ 13488$ 13500 13604 13760$ 13828 13849 13952 14297 14463$ 14465
		15144$ 17145 17148$ 17162$ 17176$ 17223 17417$ 17422 17455 17479 17579$ 17584
		17592 17651 17698 17852 17873 17956 18011
DRVST	0020	121# 2636 2646$ 2656 4756 4765$ 4784$ 4882$ 5881 5905 5920$ 5934$
		6007 6009$ 6094$ 8393 9081$ 18494 18509$ 18529$ 18608$ 19958$ 19973 19975$
		20043 20058$ 20077$ 20147 20149$ 20250$
DRVTRK	0022	128# 992 998 2305$ 4787$ 4794 4797 4811 4813$ 4820 4822$ 4825
		4895$ 4991$ 6102$ 16558 18533$ 18541 18619$ 18697$ 20253$ 21186$
DS08	FBE0	20450#
DS10	FBEA	20458# 20466
DS14	FBF1	20460 20464#
DS15	FC03	20474 20478#
DS17	FC15	20486 20491#
DS20	FC19	20495# 20503
DS22	FC21	20497 20502#
DS30	FC24	20500 20505#
DS32	FC30	20511 20516#
DSEC	0260	535# 13176$ 13393$ 13480$ 13754 14124
DSIDE	01AF	496# 8445$ 8475
DSKCNT	1C00	2525 2596 2598$ 2606 2608$ 2616 2618$ 2626 2628$ 2720 2737 2740$

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		3111	4850	4853$	4883	4885$	5371	5400	5861	5979	5994	6016	6022
		6025$	6046	6059	6062	6110	6112$	6127	18323#	18375$	18580	18583$	18610
		18612$	18999	19030	19961	19963$	20003	20158	20164	20167$	20259	20261$	20284
		20319	20322	20327	20350	20417	21268	21286					
DSKCPY	C8F0	10467#	21085	21093									
DSKFUL	=0072	15672#	17744	18075									
DSKID	0012	112#	2147$	2149$	4997	5001	5014$	5016$	5325	5327	6178	6182	11842$
		11844$	17429$	17431$	17485	17488	18706	18709	18724$	18726$	18938	18940	20547
		20551											
DSKIN1	EB76	16974#	16976										
DSKIN2	EB7E	16979#	16984										
DSKINT	EAA0	3099	16805#	21364									
DSKNAM	FE88	10307	17472	21072#									
DSKVER	0101	489#	11860$	12760	12763	17456	17481$						
DSTRT	F50A	18888	18930#	19286									
DTRCK	=0626	6087$	6153	6345	20231#	20452$	20459	20505$	20714				
DTYBAM	EF88	8764	8799	17679	17697#	17722							
DUPLC1	85A5	685	686	2284#									
DUPLCT	C8C1	10429#	10478	21081	21089								
DX0000	C90C	10477	10496#										
DX0005	C90F	10497#											
DX0010	C923	10513#	10517										
DX0020	C928	10500	10504	10515#									
DYFILE	=0040	243#	13699	14900									
E10	E718	15962#	15983										
E20	E722	15969#	15971										
E30	E727	15968	15972#										
E40	E735	15975	15981#										
E45	E739	15976	15980	15984#									
E50	E73D	15964	15988#	15990									
E55	E742	15991#	15994										
E60	E754	15992	15996	16004#									
E70	E763	16006	16014#										
E90	E74D	15986	15997#										
EA10	E76D	16023	16025#	16033									
EA20	E77E	16035	16037#										
EADV1	E767	15989	15993	16021#									
EADV2	E775	15967	15970	16032#									
ECHKSM	FEE6	21154#											
END	F99C	9195	18491	18500	18505	18534	18816	19999#	20272	20281	20844	20910	
END002	F9B1	20025#											
END003	F9C1	20029	20035#										
END10	F9D9	20046	20050#										
END20	F9E4	20054	20060#										
END30	F9FA	20051	20061	20064	20080#								

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
END3OX	FAO2	20082	20087#										
END33	FABE	20048	20084	20142	20145	20155	20168	20216#	21279				
END33X	F9D6	20041	20048#	20078									
END4O	F9CB	20026	20033	20040#									
ENDCMD	C194	4486	9307#	10692	10695	10850	10969	11072	11216	11234	11278	11288	11314
		11320	11333	11346	11789	13224	13450	13636	13649	15181	17354	17522	
ENDRD	D7EB	13222#	13240										
ENDSAV	C199	9310#	13506										
END_OF_FILE	914E	4409	4440#										
ENTFND	O253	518#	9781	9817	9837$	9891$	9968$	10019$	10181	10373	14119		
ENTIND	OODD	479#	9543$	9896$	10662	10663	10754	10839	13178$	13394	13481	14128	
ENTSEC	OOD8	478#	9542$	9898$	10666	10667	10750	10835	13175$	13392	13479	14125	
EOI	=0008	293#	920	16516									
EOIFLG	OOF8	484#	727$	1064$	10719$	10782$	11761	14836	14857	16291	16322	16370$	16661$
EOIOUT	=0080	291#	297	12062	12570	17311							
EOISND	=0008	292#	10718	10781	12500								
EPTR	O299	576#	12912$	12915	12926$	12931							
ERO	FECD	14440	14531	21129#									
EROO	FECC	15313	21128#										
ER1	FECE	14435	14550	21130#									
ER2	FECF	14534	21131#										
ER3	FEDO	14547	21132#										
ERLED	O26D	538#	9248$	9258$	12890$	13041	13059	17197					
ERMOVE	E7O6	15920	15950#	16012									
ERMSG2	E6E3	15922#											
ERR	F553	18987#	18998										
ERR1	E625	15776	15782#										
ERR1O	E698	15818	15828	15855	15857#								
ERR2	E627	15778	15784#										
ERR3	E62D	15781	15789#										
ERR4	E644	15793	15800#										
ERRBUF	O2D5	602#	12558	12561	12578	12580	15909	15911	15917	15935	16966	16969	
ERRCHN	=0005	254#	10923$	12585$	12587$	15918$	15936$	15939$	16989$	16994	17004$		
ERRMSG	E6C7	9320	15797	15803	15907#	17745							
ERROFF	C123	9245#	9321	12566	15900								
ERRON	C12C	9251#	15808										
ERROR	E6OA	1944	4502	8090	8470	9191	10458	12956	15757#				
ERRR	F969	9164	18480	18621	18720	18735	18915	18926	18987	19035	19316	19938#	20854
		20920											
ERRR1O	F975	19942	19946#										
ERRSA	=OO1O	255#	11977	16995$									
ERRTAB	E4FC	15683#	15956	15958									
ERRTSO	E6C1	12568	15902#	17046									
ERWORD	O26C	537#	2266$	9247$	9255$	9311	9436$	9539$	13634	13647	14844$	14848	15162$
		17178	17194$	17199$									

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
ESEEK	F410	18701 18723#
ETEND	=E60A	15752# 15974 15978
EX	F36E	18593 18597#
EX2	F370	18471 18598#
EX3	F379	18603#
EXBAD	838C	1937 1941# 2096 2240 2581
EXCHN2	8594	2269#
EXCHNL	8591	2268#
EXDO	93A2	4739 4863 4872#
EXE	F367	18592#
EXEC	=00E0	94# 10448
EXECD	=0060	209# 4862 5058 18592 18771
EXRD	83E8	1996 2001#
FOOO	FB43	20325# 20335 20337
FOO1	FB46	20327# 20332
FOO5	FB5C	20328 20342#
FOO6	FB64	20348# 20358 20360
FOO7	FB67	20350# 20355
FOO9	FB7D	20351 20370#
FO13	FB97	20381 20389 20392#
FO14	FB9E	20393 20398#
F1CNT	0277	550# 9402$ 9414 9424 9536$ 9556$ 9651 10706 10862 11150$ 11204 13280 13285 13573$
F1PTR	00D3	472# 9534$ 9565$ 9568 11350 11351$ 11374
F2CNT	0278	551# 4355 4358$ 4363$ 9403$ 9419$ 9537$ 9555 9558$ 9573 9652 9654$ 9655 9687 9939 10654 10737 10860 13271$ 13574$ 16209 16212$ 16224$
F2PTR	0279	552# 9415$ 9535$ 9559$ 9890 9940$ 9942$ 9946 10672$ 10708$ 10734 10744 10753 10756 11151 11184$ 13425$ 13471 13478
FAIL	8381	1934# 1981 2065 2125 3926
FASTSR	0037	146# 643 645$ 748 750$ 758 760$ 820 941 1124$ 2352 2358 2360$ 2459 2463 2465$ 4331 4333$ 4364 4366$ 4411 4413$ 4445 4606 4611 4613$ 6582 6584$ 8356 8358$ 9036 9039 9041$ 9059 9062 9064$
FB1	D2BC	12295# 12301
FB2	D2C8	12298 12304#
FB3	D2D8	12310 12313#
FBAM10	F004	17777 17778 17779 17791#
FBG10	FE44	21003# 21026
FBG15	FE64	21020 21028#
FBTOG	FE30	6217 20597 20989#
FBUFS	=0300	405#
FF10	C4AA	9795 9808# 9815
FF15	C492	9798# 9811
FF25	C4BA	9796 9806 9816#
FF30	C4C9	9818 9823#
FF40	C4D7	9809 9821 9824 9829#

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
FFRE	C48B	9794#	10385										
FFST	C49D	9803#	10355	13274	13599								
FILCNT	0295	572#	9982$	9988$	10020								
FILDRV	00E2	480#	9544$	9571$	9695	9714	9847	9918	9921$	10552	10658	10659	10701
		10745	10823	10825$	10826	10829$	10832	13098	13183$	13213$	13486	13572$	14120
		17410											
FILNOP	=0061	11774	15663#										
FILOPN	=0060	13415	15662#										
FILSEC	0285	557#	4345$	4388	9547$	9927$	10380	11013	11045	11208$	11342	11352	11375
		11378	11380	13142$	13474	16228							
FILTBL	027A	555#	4353$	8341$	9364$	9370$	9391$	9413$	9468$	9480$	9541$	9566	9569$
		9621	9656	9856	10501	10557	10558$	10845	13146	13266$	13327	13509	13575$
		13591$	17425	17474									
FILTRK	0280	556#	4374	9546$	9924$	9949	10378	10550$	10551$	10864	10873	11206$	13139$
		13314	13330	13348	13403	13472	16218	16226					
FILTYP	00EC	482#	11071$	12019	13223$	13607$	13758	13850$	13953$	14061	14066	14067$	14072
		14116	14461	15142									
FIL_SYN	9B73	6041	6134#										
FIN	F31D	18534#											
FINAL	8387	1936#	2162										
FINBAD	8384	1935#	3840										
FINDBUF	91CA	4334	4526#										
FINMFM	8C35	3579	3589#	3593	3598								
FIXBAM	EFF1	17663	17708	17774#									
FLOO	8CBA	3656	3684#										
FLO5	C6AD	10144#	10156										
FL10	C6C8	10147	10149	10154	10157#								
FLEXST	=0063	10875	13356	15665#									
FLNTFD	=0062	4517	10866	13406	15664#								
FM2030	=0042	273#											
FM4040	=0041	274#	21136										
FMHDPT	001B	117#	6263$	6281	6357	6360$	6390$	6395	6409	6411$			
FMT105	C8E0	10450#	10451										
FMT110	C8EF	10454	10459#										
FMTE10	FDDB	20908	20912#										
FMTEND	FD96	20842	20847#										
FMTERR	FDD3	20289	20340	20363	20428	20489	20514	20802	20907#				
FMTSID	01B2	499#	6096	8032$	8058$								
FMTVAR	=0620	20226#	20227	20228	20229	20230	20231	20232	20233				
FND10	D383	12461#	12465										
FND2	F1F5	18189	18192#										
FND30	D391	12462	12468#										
FNDBUF	D2BA	12259	12263	12293#									
FNDC20	D0F3	11972	11974#										
FNDC25	D0F9	11976	11978#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
FNDC30	D106	11982	11987#										
FNDFIL	C4B5	9814#	9820	9828	10281								
FNDL10	E1B7	15049#	15057										
FNDL20	E1C8	15050	15063#										
FNDL30	E1C4	15054	15059#										
FNDLMT	C6A6	9858	10096	10140#									
FNDLNX	D37F	12149	12459#										
FNDLST	E1B2	15014	15025	15045#									
FNDN0	F195	18102	18108	18111#									
FNDN1	F19A	18114#	18120										
FNDN2	F19D	18113	18116#										
FNDNXT	F173	18054	18093#										
FNDRCH	D0EB	894	10179	10713	10930	11363	11970#	12480	13376	14247	15128	15839	16487
FNDREL	CE0E	11412#	15167	15346	15490								
FNDSEC	F1E6	18156	18182#										
FNDW10	D119	12007#	12013										
FNDW13	D10F	11999	12001#										
FNDW15	D121	12005	12013#										
FNDW20	D123	12006	12014#										
FNDWCH	D107	1100	10800	11725	11997#	13669	13832	15849	16691				
FNLOOK	89A1	3034#	3041	3044									
FN_000	898A	3019#	3025										
FN_001	8996	3021	3027#										
FN_002	89B2	3036	3046#										
FN_INT	8986	2204	3017#										
FORMAT	C8C6	8609	10436#	21351									
FORMT	FAC7	10440	10442	20245#									
FOUND	028F	566#	9777	9785	9807$	9808	9819	9938$	9952$				
FRD_00	8394	1928	1947#										
FRD_01	83A3	1960#	1999										
FRD_02	83D2	1989#	1993										
FRE10	EF87	17690	17692#										
FRE20	EF7F	8773	17689#										
FRE25	D259	12191	12211#										
FREAD	=0088	87#	2391	4431	4731								
FREB1	D2F9	12342#	12348										
FREB2	D303	12345	12347#										
FREBUF	D2F3	12226	12235	12244	12338#								
FRECHN	D227	10413	12148	12181#	12354	12373	12689	12692	13623	13677	13703	15856	17396
FRECO	D22E	12183	12186#										
FREECO	C001	9212#											
FREIAC	D2DA	12318#	15234	15236									
FREICH	D4DA	9338	12687#	13625	15809								
FREMSG	C817	10337	10343#	10344									
FRERD	D22E	12187#											

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
FRERTS	EF87	17675	17693#										
FRETS	EF5F	10394	10414	11215	17665#								
FRETS2	EF62	8752	17673#										
FREUS2	EFD3	17754#											
FREUS3	EFD5	17756#	18223										
FREUSE	EFCF	8751	8786	17751#									
FREWRT	D22E	12188#											
FRI10	D2E9	12321	12329#										
FRI20	D2D9	12315#	12328	12331									
FRMERR	EA4E	16578	16713#										
FRMERX	E999	16542	16578#										
FRMT	=00F0	95#	4865										
FS10	C3B0	9660#	9663										
FS15	C3B8	9653	9661	9664#									
FS1SET	C398	9650#	10350	13594									
FSTFMT	84B7	673	674	699	700	2130#							
FSTLOAD	9080	712	4323#	9192									
FSTRD	8371	664	690	1923#									
FSTSEK	848B	670	696	2104#	2192								
FSTWRT	83EC	667	693	2007#									
FTNUM	0051	182#	6090	6103$	6120	6123	6174	6449$	6450	6463$	6480$	18412$	20245
		20254$	20543	20839$	20840	20848$	20915$	21225$					
FWAIT	FB39	20319#	20320										
FWAIT2	FB3E	20322#	20323										
FWRT_0	8402	2010	2020#										
FWRT_1	8412	2034#	2094										
FWRT_2	8415	2036#	2052										
FWRT_3	8428	2041	2045#										
FWRT_4	8420	2040#	2047										
FWRT_5	8484	2088	2096#										
FWRT_6	8487	2091	2098#										
GA1	DF9B	14567	14569#										
GA2	DFA0	14584#											
GA3	DFB0	14586	14593#										
GAFLGS	DF9E	14147	14582#	14741	14902	14917	15567						
GAP1	=000B	224#	5411	6297	19041	20663							
GAP2	=0004	225#	20508										
GAPMFM	8C4F	3559	3607#										
GAPTBL	9ACA	6032#	6033	6080									
GBERR	D20F	12167#	12174	13873	13973								
GBF1	D2A3	12260	12264	12269#	12271								
GBF2	D2B6	12268	12280#										
GBYTE	D39B	10776	12479#										
GCBYTE	CA39	10775#	13532										
GCRB10	F90C	19838#	19861										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
GCRB20	F92B	19847	19863#										
GCRBIN	F8E0	9170	18908	19806#									
GCRERR	0035	143#											
GCRFLG	0050	181#	5615$	5842	6466$	6483$	19256$	19520$	19941	20851$	20918$		
GCRHI	F8A0	19712	19718	19724	19730	19737#							
GCRL0	F8C0	19714	19720	19726	19732	19772#							
GCRPNT	0034	140#	5121$	5503$	5674	5744$	5774$	18837$	19144$	19335	19461$	19518$	19605
		19708$	19807$	19895$	20995$								
GCRTB1	A00D	5178	5204	5747	6894#								
GCRTB2	A10D	5226	5290	5753	7177#								
GCRTB3	A20D	5260	5761	7459#									
GCRTB4	A30D	5280	5767	7742#									
GCRTBA	9F0D	5192	5218	5749	6634#								
GCRTBD	9F0F	5240	5303	5755	6636#								
GCRTBE	9F1D	5256	5759	6650#									
GCRTBF	9F0E	6635#											
GCRTBG	9F2A	5276	5765	6663#									
GCRTBH	9F00	6621#											
GD1	ED7E	17310	17315#										
GE10	D433	12559	12562	12573#									
GE15	D43A	10931	12577#										
GE20	D443	10928	12576	12582#									
GE30	D445	12571	12584#										
GET	D3AA	1013	12482	12488#	16573								
GET0	D3CE	12505	12511#										
GET00	D3B4	12490	12493#										
GET1	D3D3	12501	12516#										
GET2	D3D7	12519#											
GET3	D403	12546#	12554										
GET4GB	F7E6	9173	18839	18851	19147	19168	19204	19605#	19820	19840			
GET6	D409	12517	12551#										
GETACT	DF93	10843	11365	11679	11771	11943	12025	12126	12367	12596	12674	12696	12709
		13068	13144	13763	14085	14227	14248	14295	14300	14564#	14785	15515	15584
		17122	17221	17237	17257	17274	17283	17575					
GETB1	D151	12036	12047#										
GETB2	D14D	12041	12044#										
GETBF	D1F8	12156#											
GETBUF	D28E	11639	11695	11801	12165	12173	12255#	13871	13971	17973			
GETBYT	D137	11917	11919	12034#	12056	12069	12073	12082	12574	13601	14315	14317	17304
		17390	17392										
GETD3	ED6D	17305	17307#										
GETDIR	ED67	12553	17304#										
GETERC	D414	12497	12556#										
GETHDR	DE3E	13496	13893	14004	14248#	15247	15442	16287	16324	18034			
GETINA	DFB7	11636	11693	14615#									

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
GETLNK	DE0C	14184#	15228	15437									
GETNAM	C6CE	10166#	17247										
GETPNT	D4E8	9987	10180	11293	11756	12116	12557	12696#	13713	13720	14674	15007	15028
GETPRE	D12F	11260	11344	12025#	12034	12523	14706	14941	14951	14968			
GETR2	D1E3	12142	12145#										
GETR3	D206	12163#											
GETR4	D226	12164	12172	12176#									
GETR5	D217	12166	12170#										
GETR52	D1F3	12154#											
GETR55	D1F5	12153	12155#										
GETRCH	D1E2	4529	11005	11043	12143#	13844	17213						
GETSEC	F1FA	11228	11246	18112	18119	18187	18201#						
GETSIM	CD3C	11260#	11268										
GETWCH	D1DF	12140#	13945										
GI10	DFBF	14617	14619#										
GIB20	CA52	10783	10786	10790#									
GIBYTE	CA35	10724	10772#	16283	16321								
GN05	C6F7	10182	10189#										
GN050	C6FC	10184	10191#										
GN051	C70E	10193	10200#										
GN10	C71B	10190	10207#										
GN12	C728	10211	10213#										
GN14	C73C	10217	10219	10222	10225#								
GN15	C74A	10230	10233#										
GN20	C76B	10247	10250#										
GN22	C773	10254#	10259										
GN30	C783	10263#	10270										
GN35	C793	10265	10268	10271#									
GN37	C798	10273#	10279										
GN40	C7A7	10275	10281#										
GN45	C7AB	10283#											
GNSUB	C6DE	10170	10177#										
GOTU	F33C	18542	18546	18560#									
GOTU10	F342	18564#	18567										
GO_FMT	8CDE	3688	3701	3705#									
GP1	D4EB	12698#	14488										
GP2CMD	=0009	6033#	6079										
GP2TBL	9AD3	6034#	6086										
GS10	F20D	18218#	18227										
GS20	F21D	18221	18228#										
GS30	F21F	18224	18230#										
GSSPNT	DF45	14486#	15447	15470	15475								
GTAB	0056	184#	5677$	5680$	5689$	5692	5693$	5695	5697$	5704	5705$	5709$	5714$
		5717	5718$	5720	5722$	5736$	5739	5740$	5746	5748	5752	5754	5758
		5760	5764	5766	19332$	19333$	19349$	19357$	19359$	19362	19377	19378$	19390$

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		19394	19410	19416$	19425	19426$	19438$	19440$	19442$	19445	19458	19612$	19618$
		19631	19632$	19637$	19645$	19655	19656$	19661$	19671	19672$	19678$	19685$	19700
		19701$	19705$	19711	19713	19717	19719	19723	19725	19729	19731		
GTABYE	E848	16323	16328#										
GTABYT	E839	16240	16244	16257	16264	16321#							
GTSEC	8643	2383	2390#										
HANDSK	9228	4415	4420	4442	4454	4459	4464	4473	4478	4496	4507	4598#	9067
HBID	0039	149#	6163	18415$	19897	20533							
HDRPNT	0032	138#	4786	4800	4824	4836	4909$	4911$	5053	5063	5331	5334	6119
		6124$	6157$	6159$	6164$	6167$	6171$	6175$	6179$	6183$	6187$	6189$	6198
		6204$	6221	6223$	6224$	6226$	6233$	6286	6397	18532	18545	18561	18634$
		18766	18776	18944	18947	20277	20280$	20632$	20648	20729	20732$		
HDRS	0006	111#	1966$	1968$	2068$	2070$	2297	4387$	4390$	4426	4433$	4436$	4908
		4910	8044$	8046$	8070$	8072$	8296$	8311$	8439$	8460$	8512$	8541$	12741
		12743	12789	12791	13073$	13075$	14252	14254	15769	15771	17992$	17994$	18633
HEADER	0016	114#	4983	4990	4998	5002	5013	5015	5030	5126$	5129$	5132$	5138$
		5140$	5326$	5328$	5332$	5335$	5338	5339	5340	5341	5342$	8503	8505
		8523	8526	8534$	8535$	11841	11843	18688	18696	18707	18710	18723	18725
		18742	18842$	18845$	18848$	18854$	18856$	18939$	18941$	18945$	18948$	18954	18955
		18956	18957	18959$	19900	19903	19906	19911	19914				
HEADS	8FB3	4196#	4246										
HED2TS	D552	12780	12785#	12809									
HEDOFF	D676	13001#	21322	21328									
HEXO	E69F	15871#	15876										
HEX5	E6AA	15872	15877#										
HEXDEC	E69B	15868#	15926	15931									
HINIB	003B	154#											
HOF1	D67C	13005#	13010										
HOF2	D688	13004	13012#	13017									
HOF3	D692	13003	13011	13018#									
HSKRD	85F9	1983	1991	2079	2174	2212	2220	2224	2228	2232	2236	2263	2344#
		3842	3854	3986									
HSKSND	860B	2356#											
HSKTST	85F6	2343#	2350										
IBOP	DF25	14455	14459#										
IBRD	DF1B	14430	14452#	15097	15586								
IBWT	DF21	14456#											
ICMD	0096	449#											
ICR	400D	385#	859	871$	872	885$	886	951	955	965$	1023	1071	2038
		2045	2362	3890	3898	4616	6521	6578	8624$				
ID2O	D00B	11789#											
IDLO1	EC07	17099	17108#										
IDL1	EBFF	17079	17097#	17202									
IDL10	EC81	17183	17187#										
IDL11	EC8B	17185	17193#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
IDL12	EC98	17179	17189	17195	17200#								
IDL2	EC12	17115#	17129										
IDL3	EC2B	17118	17128#										
IDL4	EC31	17131#	17137										
IDL5	EC3B	17132	17136#										
IDL6	EC58	17153	17157#										
IDL7	EC5C	17150	17161#										
IDL8	EC69	17167	17170#										
IDL9	EC6D	17164	17174#										
IDLE	EBE7	1129	17069#										
IER1	180E	330#	16934$										
IER2	1C0E	18358#	18404$	18408$									
IFR1	180D	329#	1041	1926	2008	2109$	2250	2278$	3709$	3731	3738	3745	3937
		3971	6543	6587	16637	16933$	20330	20353	21047				
IFR2	1C0D	6549	6602	18356#	18407$	21053							
IGERR	83C9	1978	1983#										
ILERR	EA4E	16470	16703	16715#									
IMAGE	028B	562#	9400$	9407	9408$	9428	9429$	9431	9620$	9631$	10498	10515	
INAC1O	FA0E	20090	20096#										
INAC2O	FA1C	20097	20105#										
INACT	FA05	18420	18422	20089#	20151	20153							
INCPNT	D1C6	12115#											
INCPTR	D1C6	10023	12114#										
INDEX	0294	571#	9895	10015	10030$	13377	13757$	13770					
INFO	001B	116#	2507$	2512$	2517$	2522$	2535$	2539$	2543$	2551$	2575$		
INITDR	D042	4224	4316	10076	11788	11831#	13251	15798	17331	17453	17517		
INITP	DCB6	13845	13909#	13947									
INNER	8B05	3309	3329#										
INQST	856B	682	2244#										
INTDRV	D005	11787#	21081	21089									
INTT1	EB4F	16950#	16958										
INTTAB	EB4B	16948#											
INTTS	F1A9	13943	18127#										
IP	0075	417#	633$	635$	649	4637$	4640$	4641	4675$	4678$	9138	9142	9150
		10978$	10981$	16849$	16851$	16854$	16855	16861$	16878$	16888	16889$	16892$	16896$
		16900	16901	16904$	16905	16906$							
IRQ	9D88	6515#	9158	21299	21301								
IRQJMP	02A9	585#	4301$	4303$	6532$	6534$	21036	21300$	21302$				
IRQ_0	9DEA	6541	6582#										
IRQ_1	9DF2	6580	6587#										
IRQ_2	9E14	6585	6609#										
IRSA	=0011	270#	10177	10685	10688$	10711	10773	12629	12687	13093	13374	13437	
ISRO1	E95C	16538#	16565										
ISRO2	E963	16544#											
ISRO3	E980	16559	16562#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
ISRO4	E987	16567# 16570
ISRCLK	E976	16552 16556#
ISRHI	E973	16549 16554#
IT2O	D024	11803 11807#
IT3O	D02C	11798 11813#
ITERR	EA4E	16714#
ITRIAL	D00E	10050 11793# 11834
ITS1	F1B8	18136# 18181
ITS2	F1CB	18142 18144 18158#
ITS3	F1DF	18167 18171#
IWSA	=0012	271# 10683 10686$ 10739 10798 11723 12637 12690
JACPTR	82C7	743 1023# 1114
JATNSRV	80CE	717# 8647
JBMP	93B0	4860 4880#
JCLEAR	9D63	3772 6040 6266 6489#
JCNVBIN	952F	4978 5110#
JDSTRT	9600	5163# 5170 5636
JEND	99CA	4729 4754 4762 4788 5103 5857# 6116 6125 6459 6477 9196
JERRR	99B5	4746 4897 5011 5025 5322 5366 5381 5405 5667 5839# 6469 6485 9165
JFMT	939F	4866 4870#
JFMTE10	9D59	6055 6475 6479#
JFMTERR	9D51	6132 6419 6474#
JFORMAT	A40D	8026# 8611 8984
JFORMT	9B29	4870 6090#
JFRMERR	8362	1018 1122#
JGCRBIN	9965	5773# 9171
JGET4GB	98D9	5123 5135 5506 5529 5563 5674# 5787 5807 9174
JILERR	8366	836 1112 1125#
JIRQ	9DDE	4300 4302 6531 6533 6572# 9159
JLCC	92BA	4715# 6600 6606 9161
JLISTEN	8342	824 1098# 1119
JMPC	=0050	207#
JOB	0045	168# 4858 4905$ 4993 5057 5074 5107 18587 18630$ 18699 18770 18787 18820
JOBN	003F	160# 4744$ 4768$ 4773$ 4829 4872 4899 5041$ 5085 5097$ 5105$ 5477 5839 18478$ 18512$ 18517$ 18552 18597 18624 18753$ 18798 18810$ 18818$ 19105 19938
JOBNUM	00F9	485# 2397 2439$ 4397 4500 4544$ 8285 8292 8307 8314 8430 8435 8456 8508 8537 9031$ 10304 11329 11339 11366$ 11816$ 11853 12618 12620 12737$ 12786 12814 12833 13101 13765$ 14249$ 14453$ 14457$ 14473 14475 17468 17641 17881 17930 17980$ 18004$
JOBRTN	0298	575# 2155$ 2438$ 8085 8086$ 8434$ 8455$ 8502$ 8986$ 10049$ 12846$ 12868 12959
JOBS	0000	82# 1971$ 2073$ 2121$ 2405 2430 2444$ 4399$ 4724 4900 5478 5480$

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		5840$	8035$	8048$	8061$	8074$	10449$	10450	11049$	12270	12273$	12433	12817$
		12855	12940	12955	12967$	12997	13045$	17131	18464	18625	19106	19108$	19939$
JREAD	9606	5108	5166#										
JREADO1	960D	4734	5167	5170#									
JSEAK	944F	4868	4953#	5482									
JSETJB	93D1	4770	4899#	5045	5106	9163							
JSETJB1	93D3	4777	4900#										
JSLOWD	A483	833	835	995	1001	4059	4113	4285	4299	8099	8101#	8621	9175
JSLOWER	A47E	991	997	2800	8097#	9177							
JSRCH	970F	5163	5325#	5409	5663								
JSYNC	9754	4956	5164	5347	5369#	6393	6421	9169					
JTALK	81EA	834	893#										
JTOP	92C8	4723#	5852										
JUMPC	=00D0	93#	4736	18467									
JVRFY	9898	5395	5626#										
JWRIGHT	976E	5168	5392#										
JWTOBIN	97F9	5475	5490#	5845									
KILL	FE00	3773	6042	6379	9166	20300	20747	20946#					
KILLP	F258	10899	11324	16208	18286#								
L213	FAF5	20246	20276#										
L214	FB00	20278	20284#										
L40	CFBF	11735	11743#										
L41	CFC9	11744	11747#										
L42	CFD8	11734	11742	11754#									
L45	CFED	11762	11764#										
L46	CFCE	11748	11750#										
L460	F432	18746	18750#										
L465	F461	18772	18778	18783#									
L470	F483	18758	18763	18768	18784	18792	18795	18803	18810#				
L480	F43A	18757#	18811										
L490	F48D	18814	18818#										
L50	CFE8	11758	11761#										
LBUSED	0257	522#	14584$	14591$	14904	14919	15571						
LCC	F2B0	6553	6560	9160	18448#	21057							
LDO1	DA62	13565#											
LDO2	DA6D	13563	13571#										
LDO3	DA86	13566	13569	13583#									
LDO5	DA90	13585	13589#										
LD10	DA9E	13581	13597#										
LD20	DAA7	13600#											
LDCMD	=000C	13558	21105#										
LEDO	=0008	351#	9221	9232	9239	16775	16789	16927	17143	17159	21124		
LED1	=0000	352#	9221	9228	9239	16775	16789	16927	17172	21124			
LEDMSK	FECA	9257	12889	21124#									
LEDOUT	=1C02	354#	21160$										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
LEDPRT	=1C00	353#	2416	2418$	2427	2429$	9222	9234$	9240	9241$	9259	9260$	13042
		13043$	13060	13061$	16776	16788	16790$	16926	16928$	17142	17181	17201$	21159$
LEDS0	C110	9226	9230#										
LEDS1	C113	9229	9233#										
LEDSON	C118	9238#											
LED_OFF	8788	2624#	2805										
LED_ON	877C	2614#	2785										
LIMIT	0276	549#	9875	10132	10157$								
LINDX	0082	429#	897	918	952	980	10168	10172$	10779	11044	11057	11438	11613
		11616	11712	11752	11755$	11984$	12009$	12018	12028	12083	12150$	12193$	12196
		12203	12218	12227	12236	12319	12366$	12483	12488	12546	13087	13170$	13222
		13372	13477	13546	13550	13602	13661$	13694	13708	13726	13750	13852	13875
		13885	13897	13909	13948	13975	13985	14015	14040	14060	14064	14071	14209
		14289	14390	14428	14486	14565	14583	14615	14636	14664	14670	14675	14681
		14715	14722	14794	14802	14973	14995	15008	15017	15030	15095	15149	15195
		15303	15350	15415	15518	15564	15568	15585	15830$	16490	16514	16544	17121$
		17218	17263	17287	17308	21342							
LINTAB	022B	506#	10685	10686$	10688$	11053	11055$	11981	12002	12155$	12189	12195$	12362
		13552$	13655	13864	13866$	13963	13966$	14104	15826	16974$	16995$	16997$	17116
LINUSE	0256	521#	10475$	12209	12210$	12461	12469	12470$	16999$				
LISNER	=0001	290#											
LISTEN	EA2E	16460	16689#	16710									
LK05	C452	9761#	9773										
LK10	C45C	9765#	9786										
LK15	C462	9766	9769#										
LK20	C470	9774#	9782										
LK25	C475	9764	9776#	9783									
LK26	C47E	9778	9781#										
LK30	C485	9775	9785#										
LOADIR	DA55	13230	13557#										
LOCK	02AF	589#	4287	4289$	4307	4309$	6525						
LONGLN	=0032	9524	15655#										
LOOKUP	C44F	4361	9760#	10653	10830	16216							
LRF	=0080	242#	10725	10788	13533	14660	14703	14829	14855	14867	14936	15134	15171
		15176	15496										
LRUEXT	CF1D	11617	11622#										
LRUILP	CEFC	11587#	11592										
LRUINT	CEFA	11585#	11619	17013									
LRULP1	CF0D	11614#	11621										
LRUTBL	00FA	486#	11589$	11595$	11614	11615$	12407						
LRUUPD	CF09	11612#	11635	11692									
LSN15	EA39	16692	16698#										
LSN30	EA44	16696	16701	16705#									
LSNACT	0079	420#	720$	752$	772$	779$	817	15820	16363$	16393$	16409$	16416$	16457
LSNADR	0077	418#	775	4259$	4267	4268$	16412	16944$					

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
LSNERR	E688	15821	15848#										
LSTBUF	0293	570#	9974$	9977	9985$								
LSTCHR	0244	508#	8976$	10923$	11059$	11271$	11284	11287$	12035	12040	12060	12084$	12525
		12537	13857$	14708$	14943	14959	15018$	15031$	15936$	17220$	17288$	17289$	17309
LSTDRV	028E	565#	4352$	9274	9324$	9563	10993	13250$	13605$				
LSTJOB	025B	534#	2443$	8286	8315$	11050$	12128	12274$	12369	12727	12818$	12884	12898
		12906$	12953$	12988$	13044	13102	14087	14298$	17124	17224$	17577		
LSTSEC	024E	514#	8831$	8833	18099$	18106	18215$	18220					
LTOO	8CA2	3644	3671#										
LWPT	001E	120#	5863	5864$	20005	20006$							
LXINT	=000F	257#	16998										
M10	CB50	10937#	10941										
M30	CB45	10914	10918	10929#									
MAIN7	8B07	3331#	3582										
MAK10	FC3F	20533#	20574										
MAPOFF	FE87	21071#											
MAPOUT	EEF4	10412	13675	14025	15489	17575#							
MASK1	=00F8	19587#	19608										
MASK2	=0007	5679	19588#	19615									
MASK2X	=00C0	5691	19589#	19627									
MASK3	=003E	19590#	19635										
MASK4	=0001	5696	19591#	19640									
MASK4X	=00F0	5703	19592#	19650									
MASK5	=000F	5708	19593#	19659									
MASK5X	=0080	5716	19594#	19666									
MASK6	=007C	19595#	19675										
MASK7	=0003	5721	19596#	19681									
MASK7X	=00E0	5738	19597#	19695									
MASK8	=001F	19598#	19704										
MAX1	F24E	18279#	18281										
MAXIM	8978	3005#	3010										
MAXMIN	8961	2203	2990#										
MAXSA	=0012	249#	506	11971	11998	14109	16973						
MAXSEC	F24B	12771	12802	17549	18098	18214	18271#						
MAXSEK	0061	195#	2230	2827	2832	2894$	3012$	3142$					
MAXTRK	02AC	587#	8027$	8281	8422	8467$	8481$	8499$	8532$	8597$	8674	8926	8946
		11243	12746	12799	17622	18064	18166	21234$	21305$				
MDIRTY	0251	516#	11837$	17585	17590$	17700$							
MDMODE	=0003	261#	13410										
MEM	CAF8	10886#	21082	21090									
MEMERR	CB4B	10888	10903	10933#									
MEMEX	CB1D	10907#											
MEMRD	CB20	10898	10909#										
MEMWRT	CB50	10901	10936#										
MFMCMD	01B0	497#	2113	2199	2214	2571	2778$	2911	2920	2933$	2951	2954	2960

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		2972$	3148$	3200	3388	3623$	3638$	3665$	3689	3712$	3749	3755$	3778$
		3984$	4064$	4118$	4137	4161$							
MFMER	8A82	2777	3218#										
MFMHDR	0024	131#	2915$	2916	2924$	2925	3172$	3186	3190	4141	4145	4155	
MFMSIZ_HI	0044	166#	3153$	3196$	3519	3794	3851	3884	3932	4015	4075		
MFMSIZ_LO	0271	543#	3152$	3194$	3534	3816	3856	3904	3958	4039	4076	4102	
MH10	D69A	13026#	13028										
MINIM	8966	2993#	2998										
MINSEK	0060	192#	2226	2820	2847$	2893	2895	3000$	3020	3028	3144$		
MINSTP	0064	200#	18426$	20096									
MISTYP	=0064	10680	13421	15139	15666#.								
MO10	EEFB	17578#											
MODE	0297	574#	13277$	13298$	13320	13334$	13409	13426	13445	13518$			
MODES	9026	4243	4274#										
MODLST	FEB2	13516	21116#	21117									
MOTOFF	8770	2604#	5885	5928	8396	9186	20066						
MOTON	8764	2594#	2644	5874	8385	9185							
MOVB1	ED5B	17295#	17299										
MOVBUF	ED59	17236	17253	17273	17294#								
MOVHED	D693	13007	13014	13021#									
MOVO10	FDF7	20936#	20940										
MOVOVR	FDF5	20604	20934#										
MOVUP	FDE5	20602	20924#	20927									
MRCNF	8A4A	3168	3177#										
MRK1	EE04	17395	17397#										
MRK2	EDEE	17388#	17399										
MSCNT	=00FF	232#	6044	6045									
MSEEK1	8A37	3165#	3169	3175									
MSEEK3	8A50	3186#											
MSG1	C80B	10337#	10340										
MSGFRE	C806	10185	10196	10335#									
MSGLEN	=000C	10336	10344#										
MTRCNT	02AB	586#	5867	5873$	5879$	8379	8384$	8390$					
MUL100	CE50	11455#	11460										
MUL200	CE57	11456	11458#										
MUL25	CE41	11443	11446#										
MUL400	CE6D	11465	11467	11469#									
MUL50	CE4C	11445	11450	11452#									
MULPLY	CE2C	11412	11436#										
MXCHNS	=0006	248#	254	256	463	464	466	468	469	470	471	482	483
		486	507	508	535	536	12325	12416	12424	12438	14590	16978	18015
MXFILS	=0005	275#	478	479	480	481	555	556	557	9476	9540	11152	
MYPA	0097	452#											
N101	EE19	17411	17416#										
N108	EE46	17427	17453#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References										
N110	EE56	17451	17458	17463#								
NAMBUF	02B1	601#	10232$	10238$	10241$	10244$	10249$	10251$	10255$	10262$	10266	10272$	10277
		10278$	10287$	10323$	10327$	10329$	10330$	10332$	10338$	17295			
NB10	E2E9	15279#	15282										
NB20	E2F1	15285#	15291										
NB30	E303	15293	15296#										
NBCMDS	=0006	11092	11117#										
NBKH	00BB	467#	8974$	11958$	13698$	13710	13729$	13823					
NBKL	00B5	465#	8973$	11956$	13696$	13709	13727	13731$	13820	13928$			
NBSIZ	=001B	277#	10285										
NBTEMP	0272	545#	10201$	10210$	10215$	10301$	10303$	15404	15407	17232	17249	17251	17269
		17271	17653$	17655$									
NCMDS	=000C	9286	21079#	21087	21095								
ND10	C7DC	10311#											
ND15	C7E5	10316#	10325										
ND20	C7ED	10313	10319	10322#									
NDBH	02FC	605#	8960$	11899$	17654	17691$	17734$	17738					
NDBL	02FA	604#	8941$	8954	8957$	17652	17689$	17732	17736$	17740			
NDKRD	837F	665	671	680	683	691	697	706	1932#				
NDKWRT	83F8	668	694	2014#									
NEW	EE0D	17409#	21086	21094									
NEWDIR	C7B7	10203	10294#	13598									
NEWMAP	EEB7	8672	8722	17528#									
NEWMPV	EEB7	17529#											
NEWSS	E44E	15451	15511#										
NEXTS	004D	178#	5038$	5064	18750$	18777							
NFCALC	D075	8740	11876#	17570									
NM10	EEC7	17540#	17568										
NM20	EED9	17553#	17559										
NM30	EEE3	17560#	17566										
NMI	FEE7	21155#	21185										
NMIFLG	0067	399#											
NMODES	=0004	13511	21117#										
NNMI	FF01	21180#	21363										
NNMI10	FF0D	21182	21186#										
NOBLK	=0065	11249	11252	15667#									
NOCFIL	=0039	15658#	16220										
NOCHNL	=0070	11008	11359	11686	11805	12168	12466	15131	15670#	17977			
NODRIV	=0074	4515	8134	9730	15674#	15780	17831						
NODRRD	Macro	69#	8128	9109	9129								
NODRV	00FF	487#	2145$	4328	21314$								
NODRWR	Macro	73#	8998	9009	9119								
NOEOI	E94B	16517	16529#	16533									
NOFILE	=0034	9387	15657#										
NOREC	=0050	13468	14979	15498	15659#	16288							

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References										
NOTFND	=0006	218#	6418	20801								
NOTLK	E915	16488	16494#									
NOTRAN	83DD	1986	1995#									
NOTRDY	=0000	288#	12511									
NOTRX	843B	2032	2055#									
NOTRXO	844D	2062	2067#									
NOTYET	D5C4	12856	12875#									
NOWRT	8479	2058	2085	2090#								
NO_IND	8AF1	3229	3312#									
NO_MAX	8980	3006	3009#									
NO_MIN	896E	2994	2997#									
NO_TRK	891A	2934#	2941									
NR	00C1	469#	13887$	14019$	14210	14671	14676	14682	14717$	14724$	14795	14804	14996
		15286$	15295$	15304	15416$							
NRBU20	E07B	14750	14766	14772#								
NRBU50	E05D	14742	14755#									
NRBU70	E06B	14757	14762#									
NRBUF	E03C	14679	14737#	14814	14949							
NS00	8CA7	3647	3674#									
NS20	E4AC	15563#										
NS40	E4D1	15583#	15603									
NS50	E4DE	15582	15589#									
NSECTK	8A7E	3140	3215#									
NSSL	=0006	245#	246	246	14538	15371						
NSSP	=0078	247#	14491									
NTYPES	=0005	9659	13520	21121#								
NULBUF	E2E2	14021	15275#	15428	15433							
NULLNK	DE19	14022	14204#	15434								
NUM	=0621	20228#	20268$	20270$	20403	20404$	20407	20408$	20891	20892		
NUMF1	D07D	11882#	11895									
NUMF2	D083	11885	11887#	11893								
NUMFRE	EF4D	10341	15348	17650#								
NUMJOB	=0006	206#	4723	4767	5040	18461	18511	18752				
NUMSEC	FED1	18283	18569	21134#								
NUMSEK	8C53	3609#	3674									
NUMSYN	=0005	223#	5430	6268	6312	19058	20639	20680				
NUM_SEC	942C	2375	4855	4943#	8689	8730	8830					
NXDB1	D4BB	12664#	12665									
NXDRBK	D48D	12644#	13124									
NXOUT	E009	14691#	15208									
NXT1	F12D	18041#	18065	18073	18078	18086						
NXT2	F15F	18060	18077#									
NXTB1	D45F	12602	12605#									
NXTBF	004E	179#	5496$	5684	5730	5779$	19137$	19622	19690	19812$		
NXTBUF	D44D	10026	10415	12596#	17398							

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References										
NXTDS	F12D	12651	18040#									
NXTERR	F15A	18058	18075#	18088								
NXTJOB	0041	162#	4776	4830$	18521	18553$						
NXTPNT	004F	180#	5493$	5686	5732	5782$	19134$	19624	19692	19815$		
NXTR15	DFE4	14666	14669#									
NXTR20	DFF6	14677	14681#									
NXTR30	E034	14693	14697	14719#								
NXTR35	E035	14714	14721#									
NXTR40	E01D	14662	14705#									
NXTR45	E018	14672	14702#									
NXTR50	E02A	14701	14713#									
NXTREC	DFD0	14656#	14967									
NXTRK	0042	163#	4779	4828$	18524	18549$	18550$					
NXTST	0062	197#	4311$	5943	6001$	6012$	6540$	18421$	18423$	20087	20100$	20102$	20113$
		20115$	20137$	20139$	20152$	20154$	20183$	20185$	20193$	20195$	20208$	20210$	
NXTTS	F11E	12097	13979	15417	15426	15512	18033#					
NZONES	FED6	8581	21137#									
OB05	CBA0	11008#	11015	11030	11033							
OB10	CBA5	11002	11011#									
OB15	CBB8	11022#	11026									
OB30	CBF1	11006	11052#									
OFFSET	FEDB	12918	12921	12932	21142#							
OK	D5C2	12858	12869	12872#								
OKERR	E6BC	9276	14852	15899#								
OKSEC	8641	2385	2389#									
ONEDRV	C312	4354	8342	9555#	13272	17409						
ONEMS	8830	2744	2753#									
ONEMS1	8834	2755#	2756	2759								
OPO2	D7CF	13209#										
OPO21	D7F3	13203	13205	13226#								
OPO4	D7FF	13229	13232#									
OPO41	D815	13227	13242#									
OPO415	D81C	13207	13246#									
OPO42	D82B	13243	13253#									
OPO49	D834	13255	13258#									
OPO5	D837	13261#										
OP10	D83C	13260	13263#									
OP100	D94A	13404	13408#									
OP110	D95C	13411	13414	13417#								
OP115	D965	13367	13369	13421#	13430							
OP120	D96A	13420	13423#									
OP125	D990	13428	13443#									
OP130	D9C3	13465	13467	13470#								
OP20	D840	13257	13264	13266#								
OP40	D876	13281	13286	13292#	13331	13335						

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
OP45	D891	13296	13306#										
OP50	D8A7	13302	13308	13315	13319#								
OP60	D8B1	13289	13326#										
OP75	D8C6	13322	13337#										
OP77	D8CD	13342#											
OP80	D8D9	13344	13348#										
OP81	D8E1	13341	13349	13351#									
OP815	D8F0	13355	13358#										
OP82	D8F5	13353	13363#										
OP90	D940	13324	13403#										
OP95	D945	13405#											
OPEN	D7B4	9189	9282	13196#									
OPF1	DA06	13494	13505#										
OPFIN	D9EF	13398	13447	13491#									
OPIR10	CA31	10766	10768#										
OPIRFL	C9FA	10673	10709	10743#									
OPNBLK	CB84	10993#	13244										
OPNIRD	D475	9975	10014	10395	10752	12625#	17387						
OPNIWR	D486	10704	12637#										
OPNRCH	DC46	12631	13219	13237	13476	13843#							
OPNTYP	D477	12627#	16232										
OPNWCH	DCDA	12639	13371	13489	13943#								
OPREAD	D9A0	10763	13444	13453#									
OPTSCH	C3CA	9683#	9760	10352	13273	13597							
OPWRIT	D9E3	13346	13350	13485#									
OR10	DC65	13854	13858#										
OR20	DC81	13872	13874#										
OR30	DCA9	13861	13895#										
ORGSA	0084	431#	790$	794	1107	9277	11739	15814	16428$	16432	16698		
OROW	DC98	13884#	14026										
OS10	C3D5	9688#	9701	9703									
OS15	C3E8	9696	9700#										
OS30	C3EF	9694	9705#										
OS35	C400	9715#	9750										
OS40	C43C	9713	9749#										
OS45	C41B	9729#	9745										
OS50	C420	9722	9732#										
OS60	C434	9719	9743#										
OS70	C439	9728	9738	9742	9746#								
OVRBUF	=0100	205#	5445	5498	5608	5639	5778	6232	6327	6424	8178	8181	8299
		8450$	8558$	8702$	8704$	8706$	8763$	8798$	18904$	19072	19139	19249	19289
		19525	19811	20695	20812	20936	21000						
OVRFLO	=0020	244#	11729	14657	14687	14807	14829	14840	14879	15134			
OW10	DCFD	13956	13961#										
OW20	DD16	13972	13974#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
P10	E2AA	15224	15226#										
P2	E289	15199	15201#										
P30	E291	15204	15207#										
P75	E2BF	15232	15235#										
P80	E2C2	15230	15238#										
PA1	1801	316#	6591	8408$	8618	8620$	16354						
PARSE	C268	8346	9181	9366	9377	9395	9417	9453#	11080	11129	13268		
PARSXQ	C146	8354	9180	9271#									
PATCH5	FF10	16808	21195#										
PATCH6	FF20	16633	21212#	21214									
PATCH7	FF2F	17439	21224#										
PATCH9	FF45	20126	21244#										
PATFLG	028A	561#	9404	9410$	9421	9464$	9469	9474$	9481	9538$	9839$	9894$	
PATTYP	00E7	481#	4370	9473$	9485$	9545$	9825	9892	9915	9917$	9947	10375	10757
		10856	13181$	13215$	13310	13338	13343	13364	13413	13417	13577$		
PB	1800	307#	733	735$	737	805	813	815$	841$	843	1035	1067	1127$
		2036	2039$	2040	2345	2346	2452	2453	3888	3891$	3893	4599	4600
		8406$	16376	16379$	16381	16443	16453	16455$	16475$	16476	16581	16583$	16588
		16590$	16595	16597$	16602	16604$	16609	16610	16664	16716$	16746	16751	16935
		17058$	17074	17076$	21212								
PBYTE	CFAF	11728	11731#										
PCMD	=0009	9296	21099#	21101									
PCR1	180C	328#	16931$										
PCR2	1C0C	4719	4721$	5424	5427$	5468	5470$	6489	6492$	9082	18354#	18390$	18455
		18457$	19052	19055$	19096	19098$	20216	20218$	20865	20868$	20946	20948$	20956
		20959$											
PD10	EA7E	16782#	16786										
PD11	EA8F	16793#	16797										
PD20	EA7F	16783#	16784										
PD21	EA90	16794#	16795										
PE20	EA74	16774#	16803										
PE30	EA75	16775#	16800										
PE40	EA8E	16791#	16802										
PEA7A	FEEA	16777	21159#										
PERR	EA71	4704	16772#	16915									
PERR2	EB1F	16862	16902	16907	16915#								
PEZRO	EA6E	16770#	16825	16832	16836								
PHASE	02FE	606#	5887	5894$	5899$	13025$	13027	13031$	20025	20032$	20037$		
PI1	DFCD	14639	14642#										
PIBYTE	CF9B	10722	10732	11722#									
POSBUF	E29C	10803	15191	15218#	15335								
POSITN	E275	15175	15190#	15494									
POTA1	180F	331#	639	867	869$	878	880$	944	946$	961	963$	2699	2702
		4197	4208	4210$	4217	4219$	4282	4284$	4296	4298$	4929	4932$	4958
		4968	5173	5184	5199	5210	5232	5248	5268	5296	5350	5413	5433

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
		5446	5457	5465	5640	5652	5955	5958	6136	6270	6283	6299	6314
		6324	6334	6347	6369	6374	6398	6425	6435	6499	6528	6530$	8275
		8334	8416	8494	8552	8570	8590	8605	8643	8665	8747	8782	8816
		8850	8882	8901	8920	8942	9018	9032	9055	21227	21247	21250	
PPPPPP	FA38	20130#	21288										
PR10	C26B	9454#	9466	9477									
PR20	C280	9461	9464#										
PR25	C283	9463	9465#										
PR28	C299	9471	9475#										
PR30	C29E	9455	9478#										
PR35	C2A0	9459	9479#										
PR40	C2B1	9483	9486#										
PRC00	86FD	2513	2517#										
PRC01	8706	2518	2522#										
PRC02	871B	2523	2527	2535#									
PRC03	8722	2536	2539#										
PRC04	8729	2540	2543#										
PRC05	8730	2544	2550#										
PRC06	873A	2552	2558#										
PRC07	875F	2576	2583#										
PRC08	8760	2579	2585#										
PRCMD	86E6	1956	2027	2111	2141	2197	2503#	9184					
PRGDRV	026E	539#	13211	13501$									
PRGSEC	026F	540#	4344	4389$	13216	13504$							
PRGTRK	007E	425#	4340	4382$	13206	13498$							
PRGTYP	=0002	267#	10677	13214	13220	13220	13246	13304					
PRSCLN	C1E5	9373#	9385	10476	13254	13584							
PS05	C160	9278	9281	9283#									
PS10	C16A	9287#	9291										
PS20	C17A	9289	9294#										
PS30	C184	9297	9299#										
PTCH10	FF82	4286	17021	21295#									
PTCH11	FF9C	17420	21314#										
PTCH12	FFA1	12923	21321#										
PTCH13	FFA6	12937	21328#										
PTCH15	FFAE	11312	21342#										
PTCH18	A5FE	8334#	16197										
PTCH19	A61C	8354#	17090										
PTCH20	A62B	8366#	19977										
PTCH21	A634	8377#	20010										
PTCH22	A65A	8405#	21198										
PTCH22R	FF18	8409	21201#										
PTCH23	A667	8416#	11848	17998									
PTCH24	A6E5	8489#	11826										
PTCH25	A738	8551#	17800										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
PTCH26	A74F	8569#	18274										
PTCH27	A764	8589#	17444										
PTCH28	A779	8605#	21235										
PTCH29	A786	8618#	17062										
PTCH30	A7B3	4621	8643#	16756	17103								
PTCH31	A7C0	8655#	16920										
PTCH32	A7C7	8665#	17337	17463									
PTCH33	A83E	8747#	17668										
PTCH34	A874	8782#	17713										
PTCH35	A8A9	8816#	18204										
PTCH36	A8DB	8850#	18044										
PTCH37	A905	8882#	18149										
PTCH38	A91E	8901#	18174										
PTCH39	A937	8920#	17615										
PTCH40	A951	8941#	11904										
PTCH41	A97F	8973#	13930										
PTCH42	A989	2156	8984#										
PTCH43	A99D	8997#	9329										
PTCH44	A9A5	9008#	10067										
PTCH45	A9AC	9018#	16759										
PTCH46	A9B9	9030#	15760										
PTCH47	A9D5	9054#	9358										
PTCH48	A9F8	9080#	18379										
PTCH49	AA02	9090#	9498	9509									
PTCH50	AA17	9108#	10082										
PTCH51	AA1D	9118#	11866										
PTCH52	AA25	9128#	17821										
PTCH53	AA2D	9138#	10985										
PU10	EAAC	16819#	16822										
PU20	EAB2	16823#	16839										
PU30	EAB7	16826#	16829										
PUPS1	C932	10547#											
PUT	CFB7	1117	11739#	16708									
PUT4BG	F6D0	19331#	19547	19578	19909	19921	21024	21028					
PUTB1	CFFD	11772	11776#	14042									
PUTBAM	F0A5	17596	17600	17880	17922#								
PUTBYT	CFF1	11303	11751	11760	11770#	12092	12101	12103	12657	12659	12664	12666	12668
		13137	13140	13143	13718	13740	13742	17226	17228	17230	17231	17233	17235
		17243	17245	17246	17250	17252	17255	17270	17272	17280	17281	17282	17296
PUTINA	DFC2	11641	11697	14636#									
PUTSS	DD8D	13991	13993	13995	13997	13999	14001	14006	14008	14039#	15457	15459	
QUE	F2F9	18498	18508#										
QUE05	F306	18517#	18539	18555									
QUE10	F301	18514#	18518										
QUE20	F320	18515	18537#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
QUERDK	8517	679	705	2192#									
QUIT	D635	12947#	12981										
QUIT2	D63F	12870	12950	12954#	12991								
R0	0086	439#	4360$	10354$	10384$	10388	10807$	13769$	13771$	13772	13779	13781$	13783
		13787	13790	13800$	13802$	13805$	13807	13810$	13814	13817$	13821$	13824$	15340$
		15387	15454$	15466	15952	15954	15957$	15959$	15963	15973	15979	15999$	16001$
		16022$	16024$	16026	16028	16034$	16036$	16214$	16248	16255$	17146$	17175	18131
		18135$	18139	18161	18164$	18184$							
R1	0087	440#	10805$	14997$	14998$	15002$	15009	15023$	15047	15338$	15365	15380	15545$
		15593	16238$	16285	16346	16348$							
R2	0088	441#	10809$	15005$	15056$	15059$	15342$	15402$	15419	15549$	15596	16241$	16250
		16266$	16269	16272$	16300$	16301							
R20	E219	15129	15133#										
R3	0089	442#	15220$	15240	15243	15250	15256	15539$	15592	16245$	16252	16274$	16298$
R30	E228	15137	15141#										
R35	E253	15159	15164#										
R4	008A	443#	15222$	15543$	15590$	15591$	15601	16258$	16276$				
R40	E253	15154	15157	15165#									
R50	E265	15169	15174#										
R60	E272	15178	15180#										
RA10	EAF2	16886#	16891	16894									
RA30	EB02	16896#	16911										
RA40	EB04	16897#	16909										
RAMEND	=0300	608#											
RAMTST	EAF0	16885#											
RBM10	F0F2	17975	17979#										
RBM20	F10A	17970	18002#										
RD0	D15D	12060#											
RD01	D164	12063#											
RD05	E15E	14938	14972#	15173	15179								
RD1	D16A	12061	12066#										
RD10	E127	14940#											
RD15	E138	14950#	15209										
RD20	E13B	14947	14952#										
RD25	E13D	14954#	14970										
RD3	D191	12057	12081#										
RD30	E14D	14960	14962#										
RD4	D192	12071	12082#										
RD40	E153	13892	14847	14944	14966#								
RDAB	DE57	10837	11666	14271#	14752	14759	14769						
RDBUF	D0C3	11915	11931	11936#	12078								
RDBYT	D156	10406	10408	12056#	12545	12604	13896						
RDGCR	83A0	1950	1958#										
RDIN	DE65	14279#											
RDLNK	DE95	13537	14313#	14739	14763								

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
RDMAX	=0006	226# 5094 18806
RDMODE	=0000	262#
RDREL	E120	12492 14935#
RDS5	DE75	14285 14288#
RDSS	DE73	13882 14287#
RDYLST	=0001	295# 296 297 13547 13957 17001
RDYTLK	=0088	294# 296 12583 13899 15938 17003 17262
REA7D	EA7D	16781# 21161
READ	=0080	86# 87 4392 4429 11936 12610 12827 14271 14279 14287 14454
READ01	F4D1	18885 18888#
READ11	F4D4	18890# 18896
READ20	F4E1	18900# 18906
READ28	F4FB	18912 18917#
READ40	F505	18920 18925#
REALUS	8065	624 641 651#
REC	0258	523# 9531$ 9928 9932$ 10548$ 10762$ 13166 13276$ 13329$ 13462 13463$ 13466 13868 13968 13996
RECO	D5F4	12902 12907#
RECO1	D5E3	12894 12896#
REC1	D600	12914# 12933
REC3	D625	12929 12934#
REC5	D631	12909 12943#
REC7	D644	8471 12945 12957#
REC8	D651	12968#
REC9	D65C	12942 12982#
REC95	D66D	12895 12960 12985 12992#
RECH	00BB	468# 11441 14667$ 15148$
RECL	00B5	466# 11439 14665$ 15146$
RECORD	E207	15124# 21083 21091
RECOV	D5C6	12867 12879#
RECOVF	=0051	14843 15161 15210 15660# 16325
RECPTR	00D4	474# 10810$ 11461 15166$ 15198 15345$
REDBAM	F0DF	8138 10296 17640 17872 17967#
REED	F4CA	18821 18884#
REL1	D26B	12221 12227#
REL10	D253	12206 12209#
REL15	D24D	12205# 12208
REL2	D27C	12230 12236#
REL3	D28D	12239 12245#
RELBUF	D25A	12167 12199 12218#
RELINX	D249	12203#
RELP05	E096	14793 14800#
RELP06	E094	14798#
RELP07	E09E	14796 14807#
RELP10	E0A3	14805 14810#

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
RELP20	E0AA	14813	14816#										
RELPTR	00D7	477#	10811$	11415$	11417$	11418$	15192	15203	15355				
RELPUT	E07C	14784#	14834	14885									
RELTYP	=0004	264#	12022	13133	13157	13288	13368	13429	13666	13776	13860	13955	15854
REMDR	=0627	20232#	20518$										
RENAME	CA88	10822#	21086	21094									
REQFST	8199	823	848#	851									
RESOFF	87AA	2638	2647#										
RESULT	008B	444#	11419	11463	11464$	11466$	11468$	11486	11489$	11528$	11530$	11532$	11541$
		11542$	11543$	11567	11569$								
RES_PNT	A496	8118#	8156	8197	8211	8271	8304	8468	8482	8964			
REVCNT	006A	402#	4189$	12908	12944	13037	17034$						
RLINDX	0271	544#											
RM10	EAC9	16847#											
RN10	CA97	10827	10829#										
RNDEOI	=0081	297#	12540	14963	14976								
RNDGET	D3DE	12509	12523#	21343									
RNDRDY	=0089	296#	11061	11272	12507	14956	15150						
RNGET1	D3EC	11277	12526	12530#									
RNGET2	D3EE	11345	11753	12532#									
RNGET3	D3FF	12538	12542#										
RNGET4	D3F0	12534#											
ROM	=C000	240#	9210										
RS	00C7	470#	11453	13869$	13969$	14018	15004	15158	15196	15309	15351		
RSTEPS	0061	196#	20108$	20189$									
RT10	EAD6	16854#	16859										
RT20	EAD8	16855#	16857										
RTCH18	E7A8	8343	8347	16203#									
RTCH19	EBFF	8359	17093#										
RTCH21	F9B1	8398	20021#										
RTCH25	F008	8562	17805#										
RTCH27	EE43	8598	17447#										
RTCH31	EB25	8658	16926#										
RTCH34	EF96	8787	17718#										
RTCH35	F1FD	8821	18209#										
RTCH36	F130	8856	18050#										
RTCH36A	F138	8872	8875	18054#									
RTCH37	F1C7	8887	18155#										
RTCH37A	F1C9	8894	18156#										
RTCH38	F1E2	8906	18180#										
RTCH38A	F1E4	8913	18181#										
RTCH43	C1B7	9000	9333#										
RTCH44	C664	9011	10070#										
RTCH46	E60D	9047	15766#										
RTCH48	F266	9083	18384#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
RTCH51	D075	9121	11870#										
RTCH52	F01B	9131	17825#										
S00	8CB5	3653	3681#										
SA	0083	430#	792$	4484$	4498$	4509$	4527$	9069$	10166	10174$	10178$	10684$	10712$
		10740$	10774$	10799$	10995	10999$	11052	11362$	11724$	11732	11970	11997	12151
		12181	12188	12353$	12355$	12360$	12361	12374$	12494	12516	12630$	12638$	12688$
		12691$	13085	13094$	13173$	13197	13294$	13375$	13435	13438$	13442$	13492	13551
		13619	13630	13642$	13645$	13654	13752	13831$	13863	13962	15127$	15816$	15825
		16430$	17211$										
SA05	C370	9622#	9630										
SA10	C383	9627	9631#										
SA20	C388	9624	9636#										
SAVBM	02AD	588#	8111$	8113$	8118	8120							
SAVPNT	002E	136#	5491$	5500$	5514$	5518$	5522$	5534$	5538$	5543$	5547$	5588$	5591$
		5596$	5602$	5775$	5785$	5794$	5798$	5802$	5812$	5817$	5821$	5825$	5830
		19132$	19141$	19155$	19159$	19163$	19173$	19177$	19182$	19186$	19229$	19232$	19237$
		19243$	19450	19514$	19523$	19533	19537	19541	19551	19556	19560	19564	19808$
		19818$	19827$	19831$	19835$	19845$	19850$	19854$	19858$	19866	19889$	19923	20991$
		20998$	21005	21009	21013	21017							
SAVSP	0049	174#	4716$	5849	18451$	19949							
SAV_PNT	A48B	8110#	8148	8187	8201	8234	8289	8426	8950				
SB10	EF07	17586	17588#										
SB20	EF24	17608#	17623										
SBM10	F022	17829	17833#										
SBM30	F03E	17842	17847	17850#									
SC15	C835	10357#	10386										
SC17	C855	10367	10373#										
SC20	C86B	10376	10384#										
SC25	C86D	10358	10362	10385#									
SC30	C872	10356	10388#										
SCAL1	DF4C	14490#	14495										
SCFLG	DD95	14058#											
SCHTBL	C440	9707	9752#										
SCR1	DDFC	14148	14152#										
SCRBAM	EEFF	17353	17507	17583#									
SCREN1	C1AD	9322#	13638	13651									
SCREND	C1A3	9318#	10392										
SCRTCH	C823	10350#	21086	21094									
SCRUB	DDF1	13680	14146#	15227	15514								
SD20	C34C	9594#	9609										
SD22	C34D	9595#	9604	9607									
SD24	C34F	9596#	9615										
SD40	C352	9588	9600#										
SD50	=C361	9591	9611#										
SDIRTY	E105	14171	14784	14900#									

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
SDR	400C	384# 856$ 954$ 1075 2049 2357$ 3902 4610$
SE10	E1D8	15082# 15088
SE20	E1DC	15081 15086#
SE30	E1EF	15092 15098#
SEAK	F3B1	18595 18656# 19110
SEARCH	C617	9774 9814 9994 10002 10008 10018# 13122
SECINC	0069	401# 4185$ 12647 12650$ 12653$ 17032$ 18095
SECKZZ	8A7A	3201 3212#
SECLES	88D1	2865 2886#
SECLMO	88C8	2867 2870 2881#
SECSEK	=0088	91# 12905
SECSS	025A	525# 13163 13459$ 13879 13983$
SECT	=0628	6160$ 6170 6208$ 6210 6364$ 6392$ 6444$ 20233# 20528$ 20539 20570$ 20572 20738$ 20772$ 20833$
SECTCV	8888	2842# 3718
SECTHI	8A76	3195 3209#
SECTLO	8A72	3193 3206#
SECTNX	886C	2820# 3872 3999
SECTOP	889E	2853# 2887
SECTOR	0081	428# 8168 8175 8832 8839$ 9318$ 9354$ 9897 9973$ 9996 10013$ 10032 10371$ 10381$ 10409$ 10751$ 10836$ 11226 11231 11240$ 11379$ 11644 11648$ 11669$ 11675$ 11818$ 11920$ 12074$ 12102 12658 12792$ 12803 13074 13089 13114 13116$ 13125 13217$ 13236$ 13388 13475$ 13503 13538 13755$ 13794$ 13808$ 13856 13880$ 13982 14000 14007 14169 14190$ 14255$ 14318$ 14472$ 15244$ 15257 15440 15443 15461$ 15548 15580 15772$ 15905$ 15930 16229$ 17347$ 17368$ 17371$ 17393$ 17505$ 17513$ 17757 17764 18071$ 18084$ 18093 18096$ 18101 18105 18107$ 18110$ 18118$ 18186$ 18219 18226$
SECTR	0043	164# 2521$ 4848$ 4852 4856$ 5033 5036 5068 6211 6391 18570$ 18745 18748 18781 20454 20496 20517 20573 20771
SECTSK	F6CA	19277 19319#
SEEK	=00B0	90# 91 2119 8513 11822
SEEK10	F3BB	18664# 18717
SEEK15	F3C8	18673# 18681
SEEK20	F407	18671 18716#
SEEK30	F3DC	18688# 18690
SEKDN	87CC	2679#
SEKDUN	87D8	2675 2685#
SEKE	87BA	2328 2491 2542 2665# 2945 2969 3108 3728 3757 3770 4131
SEKLOP	87C3	2672# 2680 2683
SEKRTN	87DD	2670 2688#
SEKTR	861E	1998 2093 2369#
SEKUP	87D2	2677 2682#
SEL_SID	8954	2550 2979# 3700
SEQGET	D400	12521 12545# 12552
SEQTYP	=0001	266# 13239 13239 13317

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
SETOO	DE2B	14164 14184 14205 14227# 15046 15276
SETANY	C368	9371 9619#
SETBAM	F011	8820 8886 8905 17752 17813# 18050
SETBPT	EF3A	8290 8551 11877 17591 17637#
SETDIR	D4EB	12697#
SETDRN	D1D3	10792 10801 11638 12076 12096 12126# 14738 15333
SETDRV	C33C	9567 9584#
SETERR	EBD5	17045#
SETFLG	DD97	10789 14060# 14704 14808 14901 15172 15497
SETH	D6D3	8029 10446 11821 12619 13069# 13881 13987 14474
SETHDR	D6DO	11914 11930 12077 12106 13068# 13946 14294 15425 15432
SETJ10	F3AB	18627 18641#
SETJB	F393	9162 18514 18624# 18757 18819
SETJB1	F395	18522 18625#
SETJOB	D50E	12735# 12837
SETLO1	E17E	15000 15003#
SETLO5	E19D	15015 15021#
SETL10	E1A4	15010 15011 15025#
SETL40	E1AC	15026 15030#
SETLDS	C100	9220# 9638 9747 9772 9801 11384 11726 12481 13218 21315
SETLE	FA4E	20136 20138 20144# 20207 20209
SETLJB	D506	11945 12726# 14302
SETLNK	DDFD	14164# 15418 15427
SETLST	E16E	14725 14995#
SETPNT	D4C8	10016 10405 10755 10770 10842 11267 11301 12068 12099 12108 12579 12633
		12655 12663 12673# 13130 13378 13545 13739 13890 14013 14314 14683 14748
		15193 15532 15559 15575 15588 17216 17389
SETSSP	DEE9	13989 14003 14403# 14439 15109 15368 15453
SET_BM	A4A1	8126# 8758 8793 8827 8867 8893 8912
SET_BUF	91DA	4384 4541#
SET_FIL	91EA	4324 4557#
SET_SIDE	93F3	2984 4840 4923# 6105 21308
SHORT	FA3B	20099 20101 20133#
SIDE	O1B1	498# 4889 4928$ 4931 6451
SIGN	8FBO	4193# 4240
SIGNATURE	924E	4193 4630# 9193
SIGNATURE_HI	8001	615# 4690
SIGNATURE_LO	8000	614# 4686
SIG_ERR	92B3	4687 4691 4702#
SIMPRS	C1D1	9363# 11787 13232 17330
SJ10	DE7F	14269 14273 14294#
SJ20	DE8B	14277 14281 14299#
SJ30	DE92	14292 14302#
SJB1	D57A	12754 12774 12813#
SJB2	D535	12758 12762#

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
SJB3	D538	12761	12764#										
SJB4	D53F	12765	12769#										
SKIP1	=0024	219#	2930	3090	3602	4592	8579						
SKIP2	=002C	220#	1931	2013	2123	2159	2388	3406	3776	4062	4116	4159	4378
		4430	4516	4893	4926	5019	5022	5317	5319	5665	6100	6455	8479
		8530	8595	9099	10315	11480	16771	18729	18732	18923	21232		
SLOWB	81AB	858#											
SLOWD	FEF3	9176	16561	21166#									
SLOWE	FEF6	21168#	21169										
SP10	C1E2	9367	9371#										
SPDCHK	9ADC	6038#	6151										
SPINP	81B2	724	862#	1128	2081	2268	3988	8355	8635				
SPOUT	81CE	853	876#	1947	2077	2171	2209	2260	3978	3982	4323		
SR1	180A	326#											
SR10	C5C4	9977#	10027										
SR15	C5CA	9978	9981#										
SR2	1C0A	18350#											
SR20	C5D7	9987#	10024										
SR30	C5FB	9991	10005#										
SR40	C629	10021	10026#										
SR50	C62F	10007	10029#										
SRCH	F510	18930	18934#	19039	19319								
SRCH20	F538	18965#	18984										
SRCH25	F53D	18969#	18978										
SRCH30	F54E	18974	18983#										
SRCHST	C5AC	9763	9805	9964#	13108	17344							
SRRE	C604	9794	10010#	17375									
SS	00CD	471#	12162$	12237	12242$	13876$	13976$	13986	14041	14290	14391	14429	14487
		15096	15519	15565	15569$	16982$							
SSA10	FA94	20177	20187#	20190	20196	20205							
SSACL	FA7B	20112	20114	20170#									
SSCALC	DF51	13693	14493#	15394									
SSDEC	FAA5	20192	20194	20198#									
SSDIR	DEDC	14377	14388#	14405	15478	15527							
SSEND	E1CB	10793	10802	13682	14434	15075#	15334						
SSIND	00D6	476#	10794	10804	10815$	11425$	13689	14438	14528	15107$	15337	15358$	15359$
		15363$	15378	15471									
SSIOFF	=0010	246#	13691	13992	14002	15080	15362	15382	15452	15523	15555	15558	
SSNUM	00D5	475#	10796	10806	10813$	11420$	13683	14525	14537	15077$	15091	15094$	15339
		15361$	15370	15386	15467								
SSP10	DF0B	14425	14433#										
SSP20	DF12	14426	14432	14437#									
SSPOS	DEF8	14423#	15168	15492									
SSRUN	FA97	20182	20184	20189#									
SSSET	DED2	14375#	14524	15076									

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References										
SSTEST	DF66	14424	14431	14523#								
ST00	8CAD	3650	3677#									
ST10	DF77	14530	14533#									
ST20	DF7B	14526	14536#									
ST30	DF8B	14539	14546#									
ST40	DF8F	14545	14549#									
STAB	0024	132#	4965$	4972$	5115	5117	5349	18662$	18670	18677$	18831	18833	18973
		19891	19894									
START8	8CBF	3658	3688#									
STATQY	8581	1935	2248	2252	2259#							
STBCTL	9FB6	2400	2420	2445	6804#	8036	8049	8062	8075	12971	13047	
STBCTR	864B	2397#	4400									
STBCTT	865E	1972	2074	2122	2413#							
STBKNT	9FB8	6806#	6807									
STBRET	8683	2404	2424	2437#								
STDIR	EC9E	13600	17210#									
STEPS	004A	175#	4780$	4781$	4888$	5897$	5946	5949	5987$	5993$	6015$	6108$	18525$
		18526$	18616$	20035$	20089	20117	20133	20157$	20257$	21278$	21284$		
STIN	87DF	2682	2693#									
STL05	D33E	12406#	12425									
STL10	D348	12409	12413#	12439								
STL20	D355	12419	12422#	12440								
STL30	D35E	12412	12421	12428#								
STL40	D363	12432#	12434									
STL50	D373	12436	12444#									
STL60	D37A	12427	12448#									
STLBUF	D339	12266	12402#									
STMTR	8794	2538	2634#									
STOUT	87E7	2679	2698#									
STP	FA69	20130	20161#									
STPFIN	8814	2696	2732#									
STPIN	FA63	20118	20157#									
STPOUT	FA32	20126#										
STR1	D0B7	11926	11928#									
STRDBL	D0AF	11923#	15245									
STRRD	D09B	11913#	11924	13851								
STRSIZ	024B	510#	10100	10141$	10150$	10153						
STRTIT	D0C9	11937	11942#									
STRTWD	884E	2784#	3059	3155	3226	3800	3942	4021	4078			
STRUCT	=FEA5	9433	21101#	21105								
SWAP	F05B	17849	17870#									
SWAP1	F0BE	17941#	17950									
SWAP2	F0D0	17925	17951#									
SWAP3	F07F	17892#	17901									
SWAP4	F09F	17908	17917#									

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
SWITCH	003B	153#	627$	1961	1977	1985	2016	2018$	2031	2056	2060	2084	2168
		2244	2247	2274	2529	2531$	2981	3228	3401	3618	3693	3697	3699$
		3760	3786	3834	3844	3885	3917	3921	3990	4126	4212	4221	4291
		4313	4367										
SYC10	FDB9	20878#	20882	20885									
SYNC	F556	9168	18664	18931	18965	18990#	20302	20774	20804				
SYNC10	F55D	18997#	19000										
SYNCLR	FDA3	20291	20865#										
SYSIRQ	FE67	9139	9143	21036#	21365								
SYSTS	=0067	15669#	18169										
SYS_ERR	91AD	4380	4504#										
SZOO	8C95	3635	3664#										
S_LOG	892A	2554	2951#										
TO	=006F	434#	8686$	8692$	8701	12403	12429$	12437	12445	12450$	17542$	17555$	17561
		17814	17835$	17866$	17871								
TOV1	C3C7	9671	9673	9675#									
T1	=0070	435#	8687$	8693$	8703	13687$	13695	14504	14505$	14594$	14596	15391$	15398
		15408	17543$	17556$	17816	17838$	17845$	17851	17864$	17879$	17888	17903	20374$
		20379$	20398$	20399$	20402	20406							
T1HC1	1805	321#	1038$	20312$	21216$								
T1HC2	1C05	4306$	5859$	6538$	18342#	18400$	20001$	20174$	20202$				
T1HL1	1807	323#	20311$										
T1HL2	1C07	4305$	5858	6537$	18345#	18399$	20000	20172	20199				
T1LC1	1804	320#	20325	20348									
T1LC2	1C04	4718	18341#	18453									
T1LL1	1806	322#	20309$										
T1LL2	1C06	18344#	18397$										
T2	=0071	436#	8688$	8694$	8705	13688$	13697	14507$	15392$	15396	17544$	17557$	20342$
		20343$	20372	20377									
T2HC1	1809	325#											
T2LC1	1808	324#											
T2LH2	1C09	18348#											
T2LL2	1C08	18347#											
T3	=0072	437#	13692$	14497	15385$								
T4	=0073	438#	13684$	13685$	14500	15388$							
TAGCMD	C1EE	9298	9384#										
TALK	E909	16469	16486#										
TALK1	E90F	16490#	16576										
TALK2	E925	16508#	16511										
TALKER	=0080	289#											
TATN20	EA6B	16747	16759#										
TBAM	029D	579#	17841	17846	17905$	17924	17929$	17960$	17962$				
TC25	C1F3	9387#	9393										
TC30	C1F8	9386	9389#	10496									
TC35	C200	9394#	13593										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
TC4O	C20A	9397	9399#										
TC5O	C228	9405	9411#										
TC6O	C245	9422	9424#										
TC7O	C24C	9425	9427#										
TC75	C254	9412	9430#										
TC8O	C26O	9434	9436#										
TEMP	006F	416#	434	435	436	437	438	2563$	2565$	2587	2734$	2739	3052
		3069$	3076	3093$	3516$	3560	4703$	8227	8239$	8253$	8261	8270$	8854
		8865	8870$	9301$	9303$	9304	9684$	9689	9692$	9697$	9698$	9702$	9904$
		9912	9913$	9916	10299$	10891$	10893$	10907	10910	10921	10922$	10924	10926
		10938$	11018$	11019$	11023$	11024$	11028	11031	11035	11038	11110$	11112$	11114
		11161$	11162$	11163$	11174	11175$	11176	11177$	11179$	11192	11200$	11205	11328$
		11331$	11335	11454$	11455$	11459	11482$	11501	11510	11526	12115$	12118	12147$
		12163$	12171$	12646$	12673$	12679	12708$	12712$	12714	13785$	13804	14103$	14107
		14113$	14118$	14122	14335$	14336$	14338$	14340$	14345	14346$	14360$	14362$	14365$
		16772	16847$	16879$	17111$	17113$	17114$	17115	17127$	17128$	17135$	17149	17163
		17680	17723	17755$	17762	18036$	18052$	18072$	18085$	18217$	18235	18238$	18249$
		18259	18263$										
TEMPSA	024C	511#	13198$	13202	13228	13293							
TGLBUF	CF8C	11661	11665	11678	11712#								
TIM	=0020	229#	18398										
TIM2	=0040	230#	4304	6536									
TIMA_H	4005	380#	8630$										
TIMA_L	4004	379#	8632$										
TIMER1	=1805	17188	17192$	18314#	18993$	18997							
TJ1O	D5BA	12861	12864	12868#									
TLERR	E68E	15831	15842	15852#									
TLKO2	E937	16506	16519#	16522									
TLKO3	E941	16524#	16527										
TLKO5	E916	16492	16497#										
TLKACT	007A	421#	721$	762$	770$	781$	827	15822	16364$	16400$	16407$	16418$	16463
TLKADR	0078	419#	766	4256$	4263	4264$	16403	16942$					
TLKERR	E680	15823	15838#										
TMP	004B	176#	4954$	5007$	5346$	5362$	6021$	6024	18657$	18716$	20163$	20166	
TN1O	C681	10106#	10109										
TN2O	C687	10102	10103	10110#									
TOBIG	=0004	216#	20488										
TOFF	029A	577#	12913$	12916	12919$	12935							
TOGDRV	C38F	9642#	9724	9733	9736	9771	9800	10195	10198				
TOLONG	=0002	214#	20339	20362									
TOMANY	=0003	215#	20427										
TOP	F2BE	18461#	19952										
TOPP	FB0C	20286	20291#	20410									
TOPRD	=0045	221#	5781	18898	19814								
TOPTST	9B89	6129	6148#										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
TOPWRT	=0045	222# 722 5443 5502 5593 5606 5638 6423 8656 15812 16366 19070 19143 19234 19247 19288 19517 20690 20806 20936 20994
TOSMAL	=0005	217# 20513
TP1LST	FEC0	10240 21122#
TP2LST	FEC5	10237 21123#
TPLST	FEB6	13525 21119#
TR10	C697	10127# 10133
TR20	C6A5	10130 10134#
TRACC	0040	161# 4837$ 5054 18562$ 18767
TRACK	0080	427# 2153$ 8150 8164 8189 8203 8229 8754 8766 8789 8801 8823 8858 8889 8908 8930 9316$ 9353$ 9971$ 10011$ 10368$ 10379$ 10389$ 10407$ 10410 10749$ 11241$ 11242 11381$ 11642 11651$ 11671$ 11677$ 11820$ 11918$ 11925 12072$ 12100 12656 12790$ 12797 13072 13091 13210$ 13234$ 13385 13473$ 13497 13762$ 13784$ 13792$ 13853 13878$ 13980 13998 14005 14166 14187$ 14253$ 14316$ 14469$ 14765 15229 15241$ 15251 15438 15445 15463$ 15544 15577 15770$ 15904$ 15925 16227$ 17349$ 17365$ 17373$ 17391$ 17394 17435$ 17502$ 17604 17607$ 17620$ 17621 17625$ 17685 17728 17840 17884 17904 18056 18062$ 18063 18069$ 18077$ 18082$ 18097 18140$ 18162$ 18213
TRAL	=0624	20230# 20432$ 20436$ 20473 20476$ 20484
TRCMBF	C688	10097 10118#
TRKNUM	FED7	18279 18564 21140#
TRKSS	0259	524# 13160 13456$ 13877 13981$
TRNAME	C66E	10095# 10847 13149 17476
TRNOFF	F98F	2569 5847 19947 19972# 21352
TRYS	=0623	6387$ 6415$ 20229# 20763$ 20798$
TSCHK	D55F	11383 12796# 17385
TSER1	D54D	12781# 12798 12800 12804 12805
TSERR	D54A	12744 12747 12773 12779#
TSTO5	FD67	20808# 20817
TSTOV1	C3BD	9623 9670# 13568
TST10	FD77	20821# 20830
TSTA50	EA63	16744 16751#
TSTATN	EA59	848 905 913 923 928 934 1007 1027 1040 1058 1086 2043 2343 2450 3896 8636 16497 16508 16519 16524 16530 16567 16623 16636 16655 16675 16743#
TSTC20	DDB9	14103# 14110
TSTC30	DDC2	14107# 14123 14126 14129
TSTC40	DDCA	14106 14113#
TSTCHN	DDB7	10357 14102#
TSTDAT	FD62	20796 20804#
TSTFLG	DDA6	10726 13534 13700 14071# 14661 14830 14841 14880 14937 15177
TSTJOB	D5A6	12842 12855#
TSTRDJ	F47E	18788 18805#
TSTRTN	EA62	16749# 16752
TSTRTS	DDC9	14111#

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
TSTTRK	003B	152#	6070$	6148	8034$	8060$							
TSTWRT	DDAB	11658	14085#	14749	14756								
TURNON	F97E	4753	18488	19956#									
TYPE	024A	509#	9532$	10759$	10858$	12628$	13096	13132$	13156	13180	13278$	13301	13305$
		13307	13312$	13318$	13366	13419	13527$	13846	13949				
TYPFIL	D125	10675	10714	10729	10765	10785	11655	11727	11743	11952	12017#	12489	12503
		12519	13663	13775	15136	15853							
TYPFLG	0296	573#	9533$	9665$	9823	9827	13247$						
TYPLST	FEBB	9660	10243	21120#	21121								
TYPMSK	=0007	269#	9826	10758	10857	13311	13365	13418					
UBAM	029B	578#	17853$	17876									
UBLKRD	CD5F	11281#	21353										
UBLKWT	CD97	11317#	21354										
UBLOCK	FFEA	651	653	21353#									
UNLSN	=003F	286#	745	16389									
UNTLK	=005F	287#	755	16396									
UNUSED	806D	655#	702	703	708	709							
UPINST	85E9	1934	1975	2078	2161	2181	2285	2333#	3779	3832	3985		
US10	CB6C	10952	10968#										
USE10	EFBA	17687	17733	17735#									
USE20	EFBD	17730	17737#										
USE30	EFB2	8808	17732#										
USEDTS	EF93	17503	17506	17710#									
USER	CB5C	10950#	21082	21090									
USERTS	EFCE	17718	17739	17742	17746#								
USREXC	CB72	10968	10971#										
USRINT	CB63	10954#	17012										
USRJMP	006B	414#	652$	654$	10977	10980							
USRTYP	=0003	268#	16231										
UTLBAD	9021	4199	4206	4230	4250	4253	4271#	4280					
UTLD00	E7C5	16219	16222#										
UTLD10	E7D8	16237#	16292										
UTLD20	E7FA	16249	16257#										
UTLD30	E802	16264#	16277										
UTLD35	E817	16273	16276#										
UTLD50	E82C	16286	16291#										
UTLODR	E7A3	16193#	21084	21092									
V1	8B23	3320	3339	3355#									
V2	8B7C	3355	3371	3385	3398	3417	3430#						
V3	8BB1	3430	3443	3458	3476#								
V4	8BEB	3476	3494	3510	3527#								
V5	8C4D	3527	3552	3570	3603#								
V6	8AF9	3239	3255	3271	3286	3301	3320#						
VAL	=0000	15792	21096#										
VALDAT	ED84	17328#											

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References											
VD10	ED9C	17346#	17376										
VD15	EDB3	17356#	17380										
VD17	EDCB	17364	17370#										
VD20	EDD4	17375#	17382										
VD25	EDD9	17345	17377#										
VERDIR	ED84	17327#	21081	21089									
VERERR	=0007	250#											
VERFMT	88F0	2911#	3742										
VERNUM	FED5	12766	17457	17480	17496	21136#							
VER_OO	8915	2922	2931#										
VER_FM	88F8	2916#	2927										
VFIN	8C48	3592	3600#										
VFMT	9CCF	6386#											
VMKBAM	EDE5	17350	17369	17374	17384#								
VNERR	D572	12767	12808#	17459									
VNMI	0065	396#	17027$	17029$	21155								
VRF10	F698	19275	19279#										
VRF15	F6A5	19289#	19297										
VRF20	F6C5	19294	19306	19315#									
VRF30	F6B3	19300#	19310										
VRFY	F691	19025	19274#										
#AITDN	8861	2805#	3060	3177	3600	3829	4065	4119					
#ATJOB	D599	2446	10422	10838	10849	11681	11916	12661	12842#	12843	13745	13883	14011
		14024	14151	14753	14760	15488	15561	15600					
#BAM	02F9	603#	8139	8143$	9273$	9309$	10400	10402$	11221$	13300$	13618$	15807$	17333$
		17352$	17776	17782$	17907	17919$	18038	18039$	18129	18130$			
#DDAT	2003	366#	3057$	3171	3243$	3259$	3275$	3290$	3305$	3324$	3343$	3359$	3375$
		3389$	3408$	3421$	3434$	3447$	3463$	3482$	3499$	3514$	3532$	3556$	3574$
		3597$	3814	3957$	4035	4092							
#DSEC	2002	365#	2918$	3798$	3936$	4019$							
#DSTAT	2000	363#	2771	2791$	2797	2811	3067	3074	3165	3236	3252	3268	3283
		3298	3317	3336	3352	3368	3382	3395	3414	3427	3440	3455	3473
		3491	3507	3524	3549	3567	3589	3806	3948	4027	4057$	4084	4111$
#DTEST	Macro	63#	2767	2787	2793	2807	3063	3070	3161	3232	3248	3264	3279
		3294	3313	3332	3348	3364	3378	3391	3410	3423	3436	3451	3469
		3487	3503	3520	3545	3563	3585	3802	3944	4023	4053	4080	4107
#DTRK	2001	364#	3056	3106$	3679$	3713$	3754$	3796$	3934$	4017$			
#FREE	EF5C	17662#											
#GP2	FD09	20716#	20721										
#HERE	A534	8149	8188	8202	8215#	8235	8951						
#LINDX	0270	541#	13373$	13391	13751$	13818							
#ORK	0044	167#	18578$	18582	18760$								
#ORKTBL	9409	4847	4936#										
#PSW	001C	119#	5876$	8387$	9118$	10045$	17010$	17011$	17152	17166			
#R10	E0B2	14832#	14869										

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References
WR20	E0B7	14835#
WR30	E0BC	14839# 14858
WR40	E0C8	14837 14842 14845#
WR45	E0D6	14849 14851#
WR50	E0D9	14831 14854#
WR51	E0E1	14859#
WR60	E0E2	14856 14861#
WRIGHT	F56E	18886 19022#
WRITE	=0090	88# 2072 11941 12615 12753 12830 12901 12949 12984 13827 14267 14275 14283 14458
WRSTAT	8596	2245 2272#
WRT0	D1A3	12093 12096# 13542
WRT05	F575	19023 19027#
WRT10	F586	19032 19037#
WRT20	F58E	19043# 19047
WRT30	F5B3	19072# 19078
WRT40	F5BF	19082# 19091
WRTAB	DE50	14267# 15560
WRTBUF	D0C7	11941# 12104 12660 13744
WRTBYT	D19D	11746 12092#
WRTC1	D0E8	11954 11957 11959#
WRTGCR	840B	2021 2029#
WRTMAX	=000C	228# 5081 18794
WRTMIN	=0009	227# 5078 18791
WRTN10	FDC9	20894# 20898 20901
WRTNUM	FDC3	20293 20298 20891#
WRTOUT	DE5E	10421 10848 14023 14150 14275# 14744 15422 15436 15599
WRTREL	E0AB	11749 14828#
WRTS10	FCB8	20641# 20645
WRTS20	FCC2	20650# 20658
WRTS30	FCD1	20665# 20672
WRTS40	FCEB	20692# 20699
WRTS50	FCF9	20704# 20711
WRTSNC	F5AB	19062# 19066
WRTSS	DE6C	14010 14283# 15487
WRTSYN	FCB1	20636# 20739
WRT_BAM	A58D	8146 8275# 17629 17786 17912
WSECT	F423	18713 18739#
WTMODE	=0001	263# 13321 13333
WTMTR	87B0	2546 2654#
WTMTR1	87B2	2656# 2658
WTOB14	F624	19168# 19190
WTOB50	F643	19179 19192#
WTOB52	F66E	19228#
WTOB53	F64F	19204# 19226

* * Cross Reference * *

Reference flags (# = Definition, $ = Write, <BLANK> = Read)

Symbol	Value	References					
WTOB55	F678	19236#	19240				
WTOB57	F685	19249#	19254				
WTOBIN	F5F2	19103	19131#	19944			
WUSED	EF90	11233	17386	17397	17707#	18115	18190
WVERFY	=00A0	89#					
XIDLE	836B	825	1128#	15861	16461	16717	
XMS	8829	2686	2744#	2746			
XXO5	F802	19621	19626#				
XXO6	F85A	19689	19694#				
ZERRES	CED9	11436	11492	11540#			
ZP2	=0065	239#	394				

This was brought to you

from the archives of

http://retro-commodore.eu