```
Addr        Line

                1       ;                       ;Printing is Condensed Pica, graphics.
            2   TITLE SIERRA/CHALLENGER OPERATING SYSTEM
            3         TAB 8   ;SET ASSEMBLER TAB STOPS
            4           NOLIST X
            5         DCARE 1
            6          FORM  44444B44444
            7   ;Options below indicated with ;<<< may be changed.  Others may not.
            8   ;
            9   CDCSYS:   EQU 1             ;<<<        ;$CDC DISK EMULATION
           10   ;
           12   NOTCDC:   EQU 0                         ;$
           13   DEFECT:   EQU 0                         ;$CDC emulation doesn't...
           14   ;                                       ;$support recorded media defects
           15   DRVNUM:   EQU 0                         ;$Don't write drive no. in header
           16   SIERRA:   EQU 0                         ;$ 0 = CHALLENGER
           23
           24   MONTH:    EQU H#08          ;<<<
           25   DAY:      EQU H#16          ;<<<
           26   YEAR:     EQU H#90          ;<<<
           27   ;
           28   REVISION: EQU H#B           ;<<<        ;REV. A: CDMF 2 and...
           29   ;                                        Swal. 5, Type K = 730 cyls.
           30   ;                                       ;REV. B: Force Log option
           31   ;                                        Cache/SSS mix.
           33   NATIVE:   EQU 1                         ;$NO EMULATION FOR CDC
           34   MULTIH:   EQU 0                         ;$NO MULTI-HOST FILE SHARING
           43
           44   SW5HARD:  EQU 1             ;<<<        ;SWALLOW-5 - HARD SECTORED
           45   ;
           47   UBS:      EQU 0                      ;$ 0 = ALLOW ILD-1 & 4 ON ALL PACKS
           52
           59   ONTARIO:  EQU 0
           62
           64   TITLE CDC/CHALLENGER O.S. -  [ASSY. NO. 56CC1]
           85
           86   ;    SOME MICROCODE FEATURES AND ENHANCEMENTS ARE INDICATED BY CERTAIN
           87   ;SYMBOLS IN THE COMMENTS:
```

Addr Line CDC/CHALLENGER O.S. - [ASSY. NO. 56CC1]

```
 96   ;$  = CDC OPERATING SYSTEM
 99   ;<  = NATIVE MODE
113
115            INCLUDE \ORG\CDC.ORG              ;$CDC Link file
116   ;This file contains the common org statements to assemble the 9200 CDC
117   ;modules.     Date: July 24, 1990
118   ;
119   BOOKLINK: EQU H#230                         ;Bookkeeping Read/Write linkage
120   BOOKKEEP: EQU H#950                         ;Bookkeeping starts here
121   SELECTA:  EQU H#B00                         ;$Selection starts here
122   OCPHAND:  EQU H#1000                        ;$OCP Handler starts here
123   MAIN:     EQU H#1800                        ;Main Code linkage starts here
124   CLKOCPA:  EQU H#1D30                        ;Start of CLKOCP module
125   DIAG:     EQU H#2100                        ;$Diagnostics start here
126   ;
127   ;
128   DIAGMAIN: EQU MAIN +     H#10     ;Linkage: Diagnostics-to-Main     ( 8 locs.)
129   SELMAIN:  EQU DIAGMAIN + H#8      ;Linkage: Selection-to-MAIN       ( 8 locs.)
130   SELMAINA: EQU SELMAIN  + H#1C     ;Linkage: Select-to-MAIN (contd.) (12 locs.)
131   CLKMAIN:  EQU SELMAINA + H#3C     ;Linkage: CLKOCP-to MAIN          (16 locs.)
132   ;
133   END                                         ;END OF 'INCLUDE' FILE
137            LIST S                    ;  ;Select Condensed Pica printing
138   TITLE2
139   LOG:      EQU NORAT                         ;Forced Error Logging (reduced retries)
140   NOLOG:    EQU RAT                           ;Normal Error Logging (& retries)
141
```

Addr Line **CDC/CHALLENGER O.S. - [ASSY. NO. 56CC1]**

```
00000              143    ORG 00
00000 E008F 3BACC  144            CTL SMC,SYSRST & CJMP NPBCLRF,CLR000 ;PROCESS SUBSYSTEM CLEAR
00001 FFFFF E02CE  145            JMP CLR00                            ;PROCESS MAINT PANEL CLEAR
                   146
                   147    ;                    WAIT LOOP
                   148
00002 3709F 5F004  149    WAIT00: RDREG0 SPIST & LDAR $+2             ;CHECK FOR E.F.
00003 E46F1 3384E  150            CTL HIO, ENCHT & BNZ EFPROC0         ;ENABLE TAGS & IF E.F., PROCESS IT
                   151    ;
00004 3709F 5F006  157    WAIT01: RDREG0   SPIST & LDAR $+2           ;E.F. pending?
00005 FFFFF 3384E  158            BNZ      EFPROC0                    ;Yes, process it
                   159    ;
                   160    ;   Seek Complete Polling
                   161    ;
00006 3714E F0054  162            MOVE MSK1,IREG & NOP SKPEND          ;IS THERE ANY SEEK...
00007 373E2 1F03B  163            RDREGB IREG,R2 & JSR SMBRD           ;COMPLETE PENDING?
                   164            BNZ.P POLL10                         ;IF YES, GO TO POLLING
00008 E0401 33BAA  164 +          SETPGE POLL10_SHR_H#B & BNZP POLL10
                   165    ;
00009 333F9 5F00F  172    WAIT02: OR ZB,B,,R9 & LDAR WAIT05            ;CHECK DELAY TIMER
0000A E7C00 3302D  174            CTL OCP,OCPDSEL & BOZ DISP00         ;IF TIME-OUT, DISPLAY ROUTINE
0000B 133F9 5F002  175            DECRB R9 & LDAR WAIT00               ;DECR. TIMER
0000C E7C00 3382D  176            CTL OCP,OCPDSEL & BNZ DISP00         ;IF NO TIMEOUT, DISPLAY ROUTINE
0000D E0090 FFFFF  177    WAIT03: CTL SMC,UMPRST                       ;DESELECT CURRENT HIO
0000E FFFFF 1F125  181    WAIT03A: JSR ALLOWALL                        ;Allow E.F's on all HAIO's
                   182    ;
                   183    WAIT05: EQU  $
0000F 3709F 5F011  187            RDREG0 SPIST & LDAR $+2              ;IF AN E.F. RECEIVED...
00010 FFFFF 3384E  188            BNZ EFPROC0                          ;PROCESS IT
                   189    ;
                   190    ;   Attention Polling
                   191    ;
00011 373F2 F002F  192            RDRAMB R2,ATTNHIO                    ;GET LAST ATTN. HAIO NUMBER
00012 34723 FFFFF  193            OR AZ,LSBZ,R2,R3                     ;COMPUTE LOCATION...
00013 01723 FFFFF  194            ADD AB,LSBZ,R2,R3                    ;OF ATTENTION PENDING...
00014 3705F F0072  195            RDMSK0 ATTN1                         ;TO CHECK
00015 0013E 1F03B  196            ADD A0,FOUTL,R3,IREG & JSR SMBRD     ;GET ATTENTION PENDING BITS
                   197            BNZ.P POLL00                         ;IF ANY, ATTENTION POLLING
00016 E0401 33B4C  197 +          SETPGE POLL00_SHR_H#B & BNZP POLL00
                   198    ;
00017 0B3F2 FFFFF  199    WAIT12: INCRB R2                             ;INCR. ATTN. HIO NUMBER
00018 45352 F0003  200            MSKRBIT R2,3                         ;IF NO. 4, RESET BACK TO ZERO
00019 3412F F002F  201            ALDRAM R2,ATTNHIO                    ;SAVE NEXT ATTN. HIO NO.
```

Addr Line **CDC/CHALLENGER O.S. - [ASSY. NO. 56CC1]**

```
                    202    ;
0001A FFFFF 1F125   203    WAIT14:   JSR ALLOWALL                    ;Allow E.F'S on all HAIO's
                    204    ;
0001B 3709F 5F01D   205              RDREGQ SPIST & LDAR WAIT15       ;IF AN E.F. RECEIVED...
0001C FFFFF 3384E   206              BNZ EFPROC0                      ;PROCESS IT
                    207    ;
                    208    ;   Clock/OCP Interrupt Processing
                    209    ;
                    210    WAIT15:   JSR.P CHKOCP                    ;CHECK FOR CLOCK OR OCP INT.
0001D E0403 1F1C9   210 +            SETPGE CHKOCP_SHR_H#B & JSRP CHKOCP
                    211    ;
                    212              JSR.P WAITSKIP                   ;Any interrupts pending?
0001E E0404 1F028   212 +            SETPGE WAITSKIP_SHR_H#B & JSRP WAITSKIP
0001F 433F9 33804   213              RESETR R9 & BNZ WAIT01           ;Yes, skip rest of Wait Loop
                    220    ;
00020 3709F 5F022   221    WAIT16:   RDREGQ SPIST & LDAR $+2          ;IF AN E.F. RECEIVED...
00021 433FD 3384E   222              RESETR R13 & BNZ EFPROC0         ;PROCESS IT
                    230    ;
                    231    ;   ILD Entry/Return
                    232    ;
00022 373F3 F00BA   233    WAIT17:   RDRAMB R3,ILDFLG                ;GET ILD FLAGS
00023 45053 F0002   234              TSTRBIT R3,2                     ;IF ANY ILD SEEK COMPLETE...
                    235              BNZ.P POLL12                     ;PENDING, GO TO POLLING
00024 E0401 33B88   235 +            SETPGE POLL12_SHR_H#B & BNZP POLL12
                    236    ;
00025 371FE F00BF   237              MOVE RAME,IREG & NOP RETURN      ;GET ILD RETURN ADDR
00026 371F9 F00BE   238              MOVE RAME,IREGH & NOP RETURNA
00027 45053 F0001   239              TSTRBIT R3,1                     ;ILD ACTIVE?
00028 C3000 63FFF   240              RD SCP,PRGCPB,R0 & CJIR NALUZ    ;YES, RETURN TO THE ILD
                    241    ;
00029 E30C0 FFFFF   242              CTL SCP,MPPGCLR                  ;RESET PUSH-BUTTON REG.
0002A 45350 F000F   243              MSKRBIT R0,H#F                   ;CHECK FOR ILD 1-F
0002B 433F9 33002   244              RESETR R9 & BOZ WAIT00           ;NO, ONCE MORE THRU WAIT LOOP
0002C FFFFF E1021   245              JMP ILD00                        ;START THE ILD
```

Addr Line **CDC/CHALLENGER O.S. - [ASSY. NO. 56CC1]**

```
                    247
                    248    ;       DISPLAY ROUTINE
                    249    ;
0002D E7C00 4F001   250    DISP00:   CTL OCP,OCPDSEL & PSHLD 1
0002E C3030 887FF   251              RD SCP,PGRMSEL,R0 & CLPCNT EVCNTZ  ;READ THUMBWHEEL SWITCHES
                    252              JSR.P FRBDISP                      ;CHECK FOR E.C. DISPLAY
0002F E0403 1F0D1   252 +            SETPGE FRBDISP_SHR_H#B & JSRP FRBDISP
00030 45050 F0400   253              TSTRBIT R0,400                     ;CHECK FOR DISPLAY < 400
                    254              BOZ.P SMDISPA                      ;IF < 400, SM-BUFFER DISPLAY
00031 E0403 3365F   254 +            SETPGE SMDISPA_SHR_H#B & BOZP SMDISPA
                    255    ;
00032 6000F FFFFF   256              XOR AQ,Q,R0                        ;MASK OFF BIT '400'
00033 327F0 FFFFF   257              OR ZQ,LSBZ,,R0                     ;MULTIPLY: THUMBWHEEL SWITCHES...
00034 0010E 1F03A   258              ADD AQ,FOUTL,R0,IREG & JSR DISP02+1 ; X 3 = SMB MEMORY ADDRESS
00035 9350F 1F039   259              WRTL SCP,LDDISP1,R0 & JSR DISP02   ;DISPLAY BITS 24-35
00036 9360F 1F039   260              WRTL SCP,LDDISP2,R0 & JSR DISP02   ;DISPLAY BITS 12-23
00037 3706F FFFFF   261              RDREGQ SMCST                       ;RESET XMC PARITY ERRORS
00038 9370F 5FFFF   262              WRTL SCP,LDDISP3,R0 & JAR          ;DISPLAY BITS 0-11 & RETURN
                    263    ;
00039 0F1EE FFFFF   264    DISP02:   INCRD IREG,IREG
0003A 37300 AF7FF   265              RDREGB RAMI,R0 & RTS               ;GET SMB DATA
                    266    ;
```

Addr Line **CDC/CHALLENGER O.S. - [ASSY. NO. 56CC1]**

```
                    268   ;    THE FOLLOWING ROUTINE WILL READ 3 CHARACTERS FROM SMB AS ADDRESSED BY IREG
                    269   ;AND ASSEMBLE THEM INTO A 36-BIT WORD IN R0 & Q.
                    270   ;
0003B 370DF 1F043   271   SMBRD:    RDREGQ RAMI & JSR SMBRD1          ;GET BITS 24-35 INTO Q
0003C 373D0 FFFFF   272            RDREGB RAMI,R0                    ;GET BITS 12-23 AND...
0003D D0FF0 1F043   273            RDXFR SMC,,R0 & JSR SMBRD1        ;MOVE THEM TO R0-UPPER
0003E 353D0 FFFFF   274            OR DB,B,RAMI,R0                   ;GET BITS 0-11 & MERGE 12-23
0003F 37351 F0020   275            RDMSKB R1,20                      ;BIT FOR XMC ERROR
00040 45111 FFFFF   276            AND DB,FOUTL,INTR1,R1             ;CHECK FOR XMC ERROR
00041 30100 A37FF   277            OR AQ,FOUTL,R0,ZERO & CRTS ALUZ   ;IF NONE, RETURN
00042 3401F E0389   278            RTOQ R1 & JMP IRPT12              ;PROCESS XMC ERROR
                    279   ;
00043 0F1EE AF7FF   280   SMBRD1:   INCRD IREG,IREG & RTS
                    281
                    282   ;    THE FOLLOWING ROUTINE WILL WRITE A 36-BIT WORD FROM Q & R0 INTO THREE
                    283   ;CONSEQUTIVE LOCATIONS OF SMB MEMORY AS ADDRESSED BY IREG.
                    284   ;
00044 321FD 1F043   285   SMBWRT:   QLDREG RAMI & JSR SMBRD1          ;SAVE BITS 24-35
00045 3490D 1F043   286            ALDREGU R0,RAMI & JSR SMBRD1      ;SAVE BITS 12-23
00046 3410D AF7FF   287            ALDREG R0,RAMI & RTS              ;SAVE BITS 0-11 & RETURN
                    288
                    290   TITLE CDC/CHALLNGER E.F. PROCESSING
                    299   ;
0004E               302   ORG H#4E
                    304
                    305   ;  THE FLAG REGISTERS ARE:
                    306   ;
                    307   ;     BIT        MEANING
                    308   ;  R14
                    309   ;       0     WRITE NEXT RECORD (WRITE)
                    310   ;       1     FORCED E.F.
                    311   ;       2     HISTORY 2 (ECCLOG)
                    315   ;_______3     HOME ADDRESS
                    316   ;       4     READ COUNT
                    317   ;             END CYL./END HEAD (ILD'S)
                    318   ;       5     SPECIAL FLAG (VARIOUS ROUTINES)
                    320   ;$      6     (not used)
                    329   ;_______7     1-TRACK
                    331   ;       8     READ HEADER MODE (READ)
                    332   ;             SEEK TO ALTERNATE (WRITE)
```

Addr Line CDC/CHALLNGER E.F. PROCESSING

```
333   ;        9      ALTERNATE TRACK
334   ;       10      U-COMPLETE
335   ;_____11        SECOND DRIVE
336   ;       12      U-BUS PARITY ERROR/U-SIDE DMA CHECK
338   ;$      13      (not used)
343   ;       14      DISK READ/WRITE STARTED
344   ;_____15        CHANNEL PARITY
345   ;       16      FIND ALTERNATE (ILD 4 & FORMAT DEFECTIVE)
346
347
348
349   ;  R15
350   ;        0      HEADER READ
351   ;        1      TRUNCATION
352   ;        2      ABSOLUTE ADDRESS (CONVERSION)
356   ;_____3         RTZ (RETURN-TO-ZERO)
358   ;$       4      DEFECTIVE (ILD)
363   ;        5      NO E.C.C. CORRECTION
367   ;        6      (NOT USED)
369   ;_____7         SEEK ONLY (SEEK)
370   ;        8      READ ONLY (SELECTION)
371   ;        9      NO DATA XFER. (READ)
372   ;               FORMAT FLAG (WRITE)
373   ;       10      SM-RETRY
374   ;_____11        ADVANCE HEAD
375   ;       12      ADVANCE CYLINDER
376   ;               COLLISION (ENDING)
377   ;       13      POLLING (SELECTION)
381   ;       14      U-INITIATED (READ & WRITE)
388   ;_____15        SEEK ERROR (SEEK ILD'S)
389   ;               DISK ECC CORRECTABLE (READ DATA SPECIAL)
```

Addr Line CDC/CHALLNGER E.F. PROCESSING

```
0004E 373F3 F002D   391    EFPROC0:  RDRAMB R3,EFALLOW                ;GET E.F. ALLOWED MASK
0004F 40333 FFFFF   392              AND AQ,B,R3,R3                   ;IS THIS HIO ALLOWED?
00050 E0084 73851   393              CTL SMC,SMCRST & CJARB NALUZ,$+1 ;IF NOT, IGNORE IT FOR NOW
                    394    ;
00051 433FF 0FFFF   395              RESETR R15 & RSTSTAK
00052 4534E F0002   396              AND DB,B,MSK1,R14 & NOP H#2      ;RESET FLAGS EXCEPT FORCED E.F.
                    397              RSTMBIT 4,ILDFLG                 ;Reset ILD Running flag
00053 373F0 F00BA   397 +            RDRAMB   R0,ILDFLG
00054 55350 F0004   397 +            RSTRBIT  R0,4
00055 3410F F00BA   397 +            ALDRAM   R0,ILDFLG
00056 7413E 5F07D   398              NALDREG R3,IREG & LDAR EFPROC3   ;BIT FOR THIS HAIO TO IREG
                    399    ;
                    400    ; SELECT THE HIO
                    401    ;
00057 3705F F0056   402              RDMSKQ $-1                       ;Q= BIAS FOR HIO SELECTION
00058 0613E FFFFF   403              ADD DQ,FOUTL,VPE,IREG            ;IREG = HIO TO BE SELECTED
00059 0F301 6F7FF   404              ADD DZC,B,ZERO,R1 & JIR          ;SELECT AN HIO
                    405    ;
0005A E4077 E005E   406              CTL HIO,SPILCH4 & JMP $+4        ;SELECT HIO 4
0005B E407B E005F   407              CTL HIO,SPILCH3 & JMP $+4        ;SELECT HIO 3
0005C E407D E0060   408              CTL HIO,SPILCH2 & JMP $+4        ;SELECT HIO 2
0005D E407E E0061   409              CTL HIO,SPILCH1 & JMP $+4        ;SELECT HIO 1
                    410    ;
0005E 337F1 FFFFF   411              OR ZB,LSBZ,,R1
0005F 337F1 FFFFF   412              OR ZB,LSBZ,,R1
00060 337F1 FFFFF   413              OR ZB,LSBZ,,R1
00061 3411F F002D   414              ALDRAM R1,EFALLOW                ;ALLOW ONLY THIS HIO
00062 3411F F002E   415              ALDRAM R1,BITHIO                 ;CURRENT HIO IS LAST HIO
00063 E4208 FFFFF   419    EFPROC1:  CTL HIO,EFREQS                   ;REQUEST THE E.F. WORD
00064 E4404 FFFFF   420              CTL HIO,HIOTGTS                  ;REQUEST THE E.F. WORD
00065 37353 F0020   421              RDMSKB R3,20                     ;SET E.F. AVAILABLE MASK
00066 45083 4F007   422              AND DB,Q,HIOST,R3 & PSHLD 7      ;E.F. AVAILABLE?
00067 3706F B386C   423              RDREGQ SMCST & CJPOP NALUZ,EFPROCB ;YES, GET E.F. WORD
00068 45083 887FF   424              AND DB,Q,HIOST,R3 & CLPCNT EVCNTZ ;E.F. AVAIL?  TIME-OUT?
00069 373FD F0205   425              RDRAMB R13,LASTDRV               ;GET LAST DRIVE NUMBER FOR...
                    426    ;
0006A E4200 FFFFF   427    EFPROCA:  CTL HIO,EFREQR                   ;NO E.F. AVAILABLE
0006B E4408 E04AD   428              CTL HIO,HIOTGTR & JMP READ62     ;EQUIPMENT CHECK
                    429    ;
0006C 373B3 1F340   430    EFPROCB:  RDREGB UBR2,R3 & JSR CLRHSSM     ;GET F.W. 12 MSB'S & CLEAR HSSM
0006D 373A0 FFFFF   431              RDREGB UBR1,R0                   ;GET F.W. 24 LSB'S
0006E 3715E F0270   432              MOVE MSK2,IREG & NOP EFSTACK     ;LOC. OF E.F. DISPLAY
                    433    ;
0006F 3705F F0003   434              RDMSKQ 3                         ;PARITY MASK
00070 46368 FFFFF   435              AND DQ,B,SMCST,R8                ;U-bus parity errors to R8
00071 E4200 FFFFF   436              CTL HIO,EFREQR                   ;CLEAR E.F. REQUEST
00072 E4408 FFFFF   437              CTL HIO,HIOTGTR
00073 34307 1F356   438              RTOR R0,R7 & JSR AGE00           ;24 LSB'S TO R7; DISPLAY AGEING
                    439    ;
00074 45357 FFFFF   440              MSKRBIT R7,H#FFFF                ;ISOLATE XFER./PREP. LENGTH
00075 4634D F003F   441              AND DQ,B,MSK1,R13 & NOP H#3F     ;ISOLATE DISK UNIT NO. IN R13
```

```
 Addr            Line CDC/CHALLNGER E.F. PROCESSING

00076 323F4 4F005   445            QTOR R4 & PSHLD 5                ;SAVE BITS 24-35 IN R4 TEMPORARILY
00077 336F0 887FF   446            LSHFTDZ R0                       ;MOVE COMMAND CODE INTO POSITION
00078 4634C F003F   447            AND DQ,B,MSK1,R12 & NOP H#3F     ;ISOLATE COMMAND CODE IN R12
00079 FFFFF E205D   448            JMP EFTRACE ;                   ;***OPTIONAL TRACE PATCH***
                    449
                    450    ; UPDATE E.F. COUNT
                    451
0007A 3715E F0287   452    EFPROCC: MOVE MSK2,IREG & NOP EFCOUNT    ;LOC. OF E.F. COUNT
0007B FFFFF 1F120   453            JSR INCCNT                       ;INCR. COUNT OF E.F'S
0007C E4408 5FFFF   454            CTL HIO,HIOTGTR & JAR            ;CONTINUE
                    455
                    456    ; CHECK FOR CHANNEL PARITY ERROR
                    457
0007D 37353 F0001   458    EFPROC3: RDMSKB R3,1                     ;CHAN. PARITY ERROR MASK
0007E 45083 FFFFF   459            AND DB,Q,HIOST,R3                ;BUS-OUT PARITY ERROR?
0007F 333F8 33907   460            OR ZB,B,,R8 & BNZ PARERR0        ;Yes, jump; U-bus parity?
00080 FFFFF 3391C   461            BNZ EFPARITY                     ;Yes, process it
                    462    ;
00081 45054 F0800   463    FWPROC0: TSTRBIT R4,800                  ;CHECK FOR BIT 35 IN FUNCT. WORD
00082 E0088 33100   464            CTL SMC,WMARST & BOZ COMREJ1      ;IF NOT BIT 35, COMMAND REJECT
00083 4505E F8000   465            TSTRBIT R14,8000                 ;CHANNEL PARITY ERROR?
                    466            BNZ.P SENS00                     ;Yes, send Sense
00084 E0401 33C51   466 +          SETPGE SENS00_SHR_H#B & BNZP SENS00
                    467    ;
00085 3710F F0242   468            RESETRAM CASHSTAT                ;RESET CACHING STATUS
00086 3710F F0302   469            RESETRAM SEEKONL                 ;Clear Seek Only flag
00087 6505C F0004   470            TSTRMSK  R12,4                   ;Sense I/O command?
00088 320FF 3308D   471            OR       ZQ,Q & BOZ FWPROC1      ;Yes, don't check drive no.
00089 2D04D F001F   472            SUBB DB,Q,MSK1,R13 & NOP D#31    ;DRIVE NO. GREATER THAN 31?
                    473            CJMP.P ALB,SEL22A                ;IF YES, INTERVENTION REQD.
0008A E0401 352F2   473 +          SETPGE SEL22A_SHR_H#B & CJMPP ALB,SEL22A
                    474    ;
0008B 373F0 F0271   475            RDRAMB R0,EFSTACK+1              ;GET E.F. BITS 12-23
0008C 45150 F0030   476            TSTBIT R0,30                     ;BIT 16 OR 17 SET?
0008D 340CF 33AAA   478    FWPROC1: RTOQ R12 & BNZ CONV34           ;$Yes, error.  Command to Q-reg.
                    491            ADD DQ,FOUTL,MSK2,IREG           ; add command and base address
0008E 0615E F1890   492          /& NOP EFCMDEC                     ; of command decoder table
                    493            OR DZ,FOUTU,MSK2,IREGH           ; set upper address
0008F 37959 F1890   494          /& NOP EFCMDEC
00090 FFFFF 6F7FF   495            JIR                              ; Jump through table
                    496
                    497    ;
                    498    TITLE  COMMAND PROCESSING - C.U. FUNCTS.
```

```
 Addr            Line  COMMAND PROCESSING - C.U. FUNCTS.


                  500   ;┌──────────────────────────────────
                  501   ;│ NO OPERATION COMMAND - 03
                  502   ;└──────────────────────────────────
                  503   NOOP:     JSR.P DOUBLE                        ;DOUBLE DRIVE NO. IF NEEDED
00091 E0403 1F125 503 +           SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
00092 3735A F0006 504   NCMPLT0: RDMSKB R10,6                         ;STATUS ACTION CODE =06
                  505   NCMPLT1: EQU $
00093 FFFFF E0D36 509             JMP END00                           ;SEND NORMAL STATUS
                  510
                  511   ;┌──────────────────────────────────
                  512   ;│ READ AND RESET DRIVE ERROR COUNTERS - 2C
                  513   ;└──────────────────────────────────
                  514   RDRSTERR: SETNBIT 90,SEN07                    ;Sense 7 = Format 9, Msg. 0
00094 37350 F0090 514 +           RDMSKB   R0,90
00095 3410F F0007 514 +           ALDRAM   R0,SEN07
                  515
                  516   ;┌──────────────────────────────────
                  517   ;│ SENSE I/O COMMAND - 04
                  518   ;└──────────────────────────────────
00096 433F0 1F366 519   SENSIO:   RESETR R0 & JSR AGESENS
00097 FFFFF E0C51 520             JMP SENS00                          ;SEND SENSE BYTES
                  521
                  530   TITLE COMMAND PROCESSING - SEEK FUNCTS.
```

```
 Addr            Line COMMAND PROCESSING - SEEK FUNCTS.


                  532   ;┌──────────────────────────────────
                  533   ;| RECALIBRATE - 13
                  534   ;└──────────────────────────────────
                  535   CALIBRAT: JSR.P SEL00                     ;CHECK DRIVE ON LINE
00098 E0401 1F169 535 +         SETPGE SEL00_SHR_H#B & JSRP SEL00
00099 433FB E0185 536           RESETR R11 & JMP SEEK20           ;RE-ZERO THE DRIVE
                  537
                  538   ;┌──────────────────────────────────
                  539   ;| TRACK POSITION RELATIVE SEEK - 0B OR
                  540   ;| SECTOR POSITION RELATIVE SEEK - 2B
                  541   ;└──────────────────────────────────
                  542   RELSEEK:  JSR.P SEL00                     ;CHECK DRIVE ON LINE
0009A E0401 1F169 542 +         SETPGE SEL00_SHR_H#B & JSRP SEL00
0009B FFFFF 1F205 543           JSR CONV01                        ;GET RELATIVE ADDRESS
0009C FFFFF E0127 544           JMP SEEK00                        ;GO SEEK
                  545
                  546   ;┌──────────────────────────────────
                  547   ;| SEEK - 07 OR SEEK AND SET SECTOR - 27
                  548   ;└──────────────────────────────────
                  549   SEEK:     JSR.P SEL00                     ;CHECK DRIVE ON LINE
0009D E0401 1F169 549 +         SETPGE SEL00_SHR_H#B & JSRP SEL00
0009E FFFFF 1F204 550           JSR CONV00                        ;GET ABSOLUTE ADDRESS
0009F FFFFF E0127 551           JMP SEEK00                        ;GO SEEK
                  552
                  553   TITLE COMMAND PROCESSING - WRITE FUNCTS.
```

```
 Addr            Line COMMAND PROCESSING - WRITE FUNCTS.


                  555   ;┌──────────────────────────────────
                  556   ;│ WRITE DATA AND PAD ZEROS - 09
                  557   ;└──────────────────────────────────
000A0 3535F F0010 558   WRANDPAD: SETRBIT R15,10                  ;SET NO READ LAST RECORD FLAG
                  559
                  560   ;┌──────────────────────────────────
                  561   ;│ WRITE DATA - 0D
                  562   ;└──────────────────────────────────
                  563   WRDATA:   JSR.P SEL00                     ;CHECK DRIVE ON LINE
000A1 E0401 1F169 563 +         SETPGE SEL00_SHR_H#B & JSRP SEL00
000A2 FFFFF 3311A 564           BOZ XFER00                        ;IF XFER. LENGTH =0, ERROR
                  565
000A3 4505F F0100 566           TSTRBIT R15,100                   ;CHECK FOR READ ONLY
000A4 FFFFF 33914 567           BNZ RDONLY                        ;IF READ ONLY, ERROR
000A5 FFFFF 1F0B2 568           JSR CONVCHK                       ;GET & CHECK RELATIVE ADDRESS
000A6 FFFFF E053C 569           JMP WRT000                        ;GO WRITE
                  570
                  571   ;┌──────────────────────────────────
                  572   ;│ PREP TRACK - 01
                  573   ;└──────────────────────────────────
000A7 3535E F0080 574   PREPTRAK: SETRBIT R14,80                  ;SET 1-TRACK FLAG
                  575
                  576   ;┌──────────────────────────────────
                  577   ;│ PREP CYLINDER - 21
                  578   ;└──────────────────────────────────
                  579   PREPCYL:  JSR.P SEL00                     ;CHECK DRIVE ON-LINE
000A8 E0401 1F169 579 +         SETPGE SEL00_SHR_H#B & JSRP SEL00
000A9 3714E F00AE 580           MOVE MSK1,IREG & NOP PREPA        ;JUMP LOC. TO IREG
000AA 65057 F0074 582           TSTRMSK R7,D#116                  ;$PREP. LENGTH = 116?
000AB FFFFF 637FF 583           CJIR ALUZ                         ;$YES, CONTINUE
000AC 320FF FFFFF 584           OR ZQ,Q
000AD FFFFF 33918 590           BNZ FMTBAD0                       ;IF NOT, ERROR
000AE 4505F F0100 591   PREPA:   TSTRBIT R15,100                  ;CHECK FOR READ ONLY
000AF FFFFF 33914 592           BNZ RDONLY                        ;IF READ ONLY, ERROR
000B0 FFFFF 1F204 593           JSR CONV00                        ;GET ABSOLUTE ADDRESS
000B1 FFFFF E1F34 594           JMP PREP00                        ;GO FORMAT WITH CONTROLLER DATA
                  595
                  612   TITLE COMMAND PROCESSING - READ FUNCTS.
```

Addr Line COMMAND PROCESSING - READ FUNCTS.

```
000B2 FFFFF 1F205   647   CONVCHK:  JSR CONV01                    ;GET RELATIVE ADDRESS
000B3 FFFFF E1DB7   649             JMP ADDRCHK                   ;$Check relative address
                    655
                    656   ;┌──────────────────────────────────
                    657   ;| READ DATA SPECIAL - 3E
                    658   ;└──────────────────────────────────
000B4 3535F F0020   659   RDDATSPC: SETRBIT R15,20                ;SET FLAG: NO ECC CORRECTION
000B5 FFFFF E00B7   660             JMP READDATA                  ;NORMAL READ
                    661
                    662   ;┌──────────────────────────────────
                    663   ;| READ VERIFY  - 0A
                    664   ;└──────────────────────────────────
                    665   RDVERIFY: JSR.P SETVERFY                ;SET VERIFY MODE & SEEK ONLY FLAG
000B6 E0403 1F233   665 +           SETPGE SETVERFY_SHR_H#B & JSRP SETVERFY
                    666
                    667   ;┌──────────────────────────────────
                    668   ;| READ DATA - 0E
                    669   ;└──────────────────────────────────
                    670   READDATA: JSR.P SEL00                   ;CHECK DRIVE ON LINE
000B7 E0401 1F169   670 +           SETPGE SEL00_SHR_H#B & JSRP SEL00
000B8 FFFFF 3311A   671             BOZ XFER00                    ;IF TRANSFER LENGTH = 0, ERROR
000B9 FFFFF 1F0B2   672             JSR CONVCHK                   ;GET & CHECK RELATIVE ADDRESS
000BA FFFFF E0407   673             JMP READ00                    ;READ THE DATA
                    674
                    675   ;┌──────────────────────────────────
                    676   ;| READ FULL TRACK - 16
                    677   ;└──────────────────────────────────
000BB 3535E F0088   678   RDTRACK:  SETRBIT R14,8B                ;SET FLAGS: HOME ADDR. & 1-TRACK
                    679   RDTRAK1:  JSR.P SEL00                   ;CHECK DRIVE ON LINE
000BC E0401 1F169   679 +           SETPGE SEL00_SHR_H#B & JSRP SEL00
000BD FFFFF 3311A   680             BOZ XFER00                    ;IF XFER LENGTH =0, ERROR
000BE FFFFF 1F204   681   RDTRAK2:  JSR CONV00                    ;GET & CONVERT ABSOLUTE ADDR.
000BF 433F6 E0407   682             RESETR R6 & JMP READ00        ;GO READ
                    683
                    684   ;┌──────────────────────────────────
                    685   ;| READ HOME ADDRESS SPECIAL - 36
                    686   ;└──────────────────────────────────
000C0 3535E F0008   687   RDSPECAL: SETRBIT R14,8                 ;SET HOME ADDRESS FLAG
000C1 37357 F0004   688             RDMSKB R7,4                   ;SET TRANSFER LENGTH = 4
000C2 FFFFF E00BC   689             JMP RDTRAK1                   ;CONTINUE
                    690
                    691   ;┌──────────────────────────────────
                    692   ;| READ COUNT - 12
                    693   ;└──────────────────────────────────
000C3 3535E F0110   694   READCNT:  SETRBIT R14,110               ;SET READ HEADER & COUNT FLAGS
000C4 37357 F0002   695             RDMSKB R7,2                   ;SET TRANSFER LENGTH =2
```

```
 Addr            Line COMMAND PROCESSING - READ FUNCTS.

                  696            JSR.P SEL00                        ;CHECK DRIVE ON LINE
000C5 E0401 1F169 696 +          SETPGE SEL00_SHR_H#B & JSRP SEL00
000C6 3437A 1F1E9 697            RTOR R7,R10 & JSR CYL00            ;COMPUTE LOC. OF CUR. CYLINDER
000C7 373DB FFFFF 698            RDREGB RAMI,R11                    ;GET CURRENT CYLINDER
                  699            JSR.P CHK2NDA                      ;CHECK FOR A SECOND DRIVE
000CB E0403 1F192 699 +          SETPGE CHK2NDA_SHR_H#B & JSRP CHK2NDA
000C9 373F9 F0015 700            RDRAMB R9,SEN412                   ;GET CURRENT RECORD NO.
000CA 433F6 1F1F1 701            RESETR R6 & JSR HEADRD             ;GET CURRENT HEAD THIS DRIVE,
000CB 3432C FFFFF 702            RTOR R2,R12                        ;MOVE IT TO R12...
000CC 433F5 E040A 703            RESETR R5 & JMP READ001            ;AND GO READ THE COUNT
                  704
                  715   ;┌──────────────────────────────────
                  716   ;| SPACE COUNT AND READ DATA - 26
                  717   ;└──────────────────────────────────
000CD 3535E F0080 718   SPACE.RD: SETRBIT R14,80                    ;SET 1-TRACK FLAG
000CE 3535F F0020 719            SETRBIT R15,20                     ;SET NO ECC CORRECTION FLAG
                  720            JSR.P SEL00                        ;CHECK DRIVE ON-LINE
000CF E0401 1F169 720 +          SETPGE SEL00_SHR_H#B & JSRP SEL00
000D0 373F0 F02CD 721            RDRAMB R0,RECLNTH                  ;GET RECORD LENGTH AND...
000D1 FFFFF 1F368 722            JSR FOUR5DN                        ;CONVERT IT TO BYTE LENGTH
000D2 61170 FFFFF 723            XOR AB,FOUTL,R7,R0                 ;IF BYTE RECORD LENGTH NOT...
000D3 FFFFF 33AAB 724            BNZ CONV35                         ;CORRECT, COMMAND REJECT
000D4 FFFFF E00BE 725            JMP RDTRAK2                        ;GO READ
                  726
                  727   TITLE  COMMAND PROCESSING - ERRORS
```

Addr Line COMMAND PROCESSING - ERRORS

```
00100              729    ORG H#100
                   730    ;┌──────────────────────────────
                   731    ;│ COMMAND REJECT ROUTINE
                   732    ;└──────────────────────────────
00100 37352 F0001  733    COMREJ1:  RDMSKB R2,1                    ;SET SENSE 7: INVALID COMMAND
                   734              JSR.P DOUBLE                   ;DOUBLE DRIVE NO. IF NEEDED
00101 E0403 1F125  734 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
00102 37353 F0004  735    COMREJ1A: RDMSKB R3,4                    ;RECOV. ACTION = 4 (COMM. REJ.)
00103 3412F F0007  736    COMREJ2:  ALDRAM R2,SEN07                ;SAVE SENSE BYTE 7
                   737              SETMBIT 80,SEN00               ;Sense 0 = Command Reject
00104 37350 F0080  737 +            RDMSKB   R0,80
00105 3410F F0000  737 +            ALDRAM   R0,SEN00
00106 FFFFF E0D3F  738              JMP END10                      ;SEND UNIT CHECK
                   739    ;
                   740    ; CHANNEL PARITY ERROR
                   741    ;
00107 3535E F8000  742    PARERR0:  SETRBIT R14,8000               ;SET CHANNEL PARITY FLAG
00108 6505C F0004  743              TSTRMSK R12,4                  ;CHECK FOR SENSE I/O COMMAND
00109 FFFFF 33081  744              BOZ FWPROC0                    ;IF SENSE I/O, PROCESS IT
                   745              JSR.P DOUBLE                   ;DOUBLE DRIVE NO. IF NEEDED
0010A E0403 1F125  745 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
                   746    ;
0010B 37359 F000D  747    PARERR1:  RDMSKB R9,H#D                  ;STATUS ACTION = 0D
0010C FFFFF E010E  748              JMP $+2
0010D 37359 F000B  749    PARERR1A: RDMSKB R9,H#B                  ;STATUS ACTION = 0B
                   750              SETMBIT 20,SEN00               ;SENSE 0 = BUS OUT PARITY ERROR
0010E 37350 F0020  750 +            RDMSKB   R0,20
0010F 3410F F0000  750 +            ALDRAM   R0,SEN00
00110 37353 F0007  751              RDMSKB R3,7                    ;RECOVERY ACTION = 7
00111 FFFFF E0D40  752              JMP END10A                     ;SEND UNIT CHECK
                   753
                   754    ; READ ONLY SWITCH ON DRIVE
                   755    ;
00112 37353 F0020  756    RDONLYA:  RDMSKB R3,20                   ;SET SENSE 12, FORMAT 1 =...
00113 3413F F000C  757              ALDRAM R3,SEN112               ;WRITE READY UNSAFE
                   758    RDONLY:   SETMBIT 2,SEN01                ;Set Sense 1: Write Inhibited
00114 37350 F0002  758 +            RDMSKB   R0,2
00115 3410F F0001  758 +            ALDRAM   R0,SEN01
00116 37352 F0010  759              RDMSKB R2,10                   ;SET SENSE 7: FORMAT 1, MSG. 0
00117 0F303 E0103  760              ADD DZC,B,ZERO,R3 & JMP COMREJ2 ;RECOVERY ACTION = 1
                   761
                   762    ; BAD FORMATTING INFORMATION
                   763    ;
00118 37352 F0008  764    FMTBAD0:  RDMSKB R2,8                    ;SENSE 7: WRONG PREP. LENGTH
00119 FFFFF E0102  765              JMP COMREJ1A
                   766
                   767    ; DATA TRANSFER LENGTH =0
                   768    ;
0011A 37352 F0003  769    XFER00:   RDMSKB R2,3                    ;SET SENSE 7: WORD COUNT LESS THAN
```

```
 Addr            Line  COMMAND PROCESSING - ERRORS

0011B FFFFF E0102  770           JMP COMREJ1A                    ;REQUIRED
                   771
                   772   ;    Process U-bus Parity error on a Function word.
                   773   ;
0011C 6505C F0004  774   EFPARITY: TSTRMSK R12,4                  ;Sense I/O command?
0011D FFFFF 33081  775           BOZ FWPROC0                     ;Yes, continue processing
                   776           JSR.P DOUBLE                    ;Double drive no. if needed
0011E E0403 1F125  776 +         SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
0011F 34383 E03BA  777           RTOR R8,R3 & JMP IRPT12+1       ;Send Equipment Check
                   778
                   779
                   780   ;    THIS SUBROUTINE INCREMENTS A 36-BIT COUNT IN 3 CONSEQUTIVE SMB MEMORY
                   781   ;LOCATIONS.
                   782   ;UPON ENTRY:  IREG = LOC. OF 12 LSB'S TO BE INCREMENTED
                   783   ;REGISTERS USED:  Q,IREG & EVCNTR
                   784   ;
00120 370DF C0002  785   INCCNT:   RDREGQ RAMI & LDCT 2           ;GET 12 LSB'S AND...
00121 0A1FD FFFFF  786           ADD ZQC,FOUTL,,RAMI             ;INCREMENT THEM
00122 271EE A6FFF  787           DECRD IREG,IREG & CRTS NLCZ     ;IF NO OVERFLOW, RETURN
00123 370DF 98121  788           RDREGQ RAMI & CJPCNT EVCNTZ,INCCNT+1 ;GET NEXT 12 BITS
00124 FFFFF AF7FF  789           RTS                             ;RETURN IF 36-BIT COUNT OVERFLOW
                   790
                   791   ;    This subroutine allows E.F's on all HAIO's.
                   792   ;
00125 37350 F000F  793   ALLOWALL: RDMSKB R0,H#F                  ;Bits to allow all E.F's
00126 3410F AF02D  794           ALDREG R0,RAME & RTS EFALLOW    ;Set them and return
                   795
                   797   TITLE CDC O.S. - SEEK ROUTINE
```

Addr Line CDC O.S. -- SEEK ROUTINE

```
                    806
                    807
                    808   ;     THE SEEK ROUTINE WILL SET HEAD (TAG 2) ON THE SELECTED DRIVE.
                    809   ;THEN THE ACTUAL SEEK WILL BE STARTED. (SET CYLINDER - TAG 1)
                    810   ;     THE HEAD NO. IS TAKEN FROM SENSE BYTE 11, FORMAT 4,5.
                    811   ;     THE CYLINDER NO. IS TAKEN FROM SENSE BYTES 8 & 9, FORMAT 4,5.
                    812   ;
                    813   ;     THE "SEEK ONLY" FLAG INDICATES A SEEK FUNCTION.  THE ABSENCE OF THIS FLAG
                    814   ;INDICATES A SEEK INITIATED AS A RESULT OF A READ OR WRITE FUNCTION.
                    815   ;     THE "RTZ" FLAG = RETURN TO ZERO (RE-ZERO).
                    816   ;
                    817   ;    REGISTERS:
                    818   ;       R0,R1,R2,R3,R8,Q & IREG = SCRATCH
                    819   ;       R4 = BIT FOR THIS DRIVE (0-23)
                    820   ;       R5-R7,R9-R10 = RESERVED FOR READ/WRITE
                    821   ;       R11 = DESIRED CYLINDER
                    822   ;       R12 = RESERVED FOR READ/WRITE
                    823   ;       R13 = PHYSICAL DRIVE NO.
                    824   ;       R14,R15 = FLAG REGISTERS
                    825   ;
                    826
                    827   ;    DRIVE STATUS EQUATES
                    828
                    829   ;READY:    EQU H#01          ;DRIVE READY
                    830   ;ONCYL:    EQU H#02          ;DRIVE ON CYLINDER
                    831   ;SEEKERR:  EQU H#04          ;DRIVE SEEK ERROR
                    832   ;FAULT:    EQU H#08          ;DRIVE FAULT
                    833   ;WRITPROT: EQU H#10          ;DRIVE WRITE PROTECTED
                    834   ;SELECT:   EQU H#20          ;DRIVE SELECTED
                    835   ;BUSY:     EQU H#40          ;DRIVE BUSY
                    836   ;SKEND:    EQU H#80          ;SEEK END
                    837
00127 5535F F0008   838   SEEK00:   RSTRBIT R15,8                  ;RESET RTZ FLAG
00128 3535F F0080   839             SETRBIT R15,80                 ;SET SEEK ONLY FLAG
00129 FFFFF E012B   844             JMP $+2
0012A 5535F F0008   845   SEEK02:   RSTRBIT R15,8                  ;RESET RTZ FLAG
0012B 4505E F0020   847             TSTRBIT  R14,20                ;$Special flag set?
0012C 603EE 3312E   852             XOR AQ,B,R14,R14 & BOZ $+2     ;**YES, RESET IT; NO, JUMP
0012D 35355 F8000   853             SETRBIT R5,8000                ;**SET DEFECT TRACK FLAG
                    854   ;
0012E 4505E F0200   855             TSTRBIT R14,200                ;ALT. TRACK FLAG?
0012F E0082 33133   856             CTL SMC,HSSMRST &  BOZ $+4     ;IF NOT, JUMP
00130 373F1 F00B0   857             RDRAMB R1,ALTHEAD              ;GET THE ALTERNATE HEAD...
00131 373FB F00AF   858             RDRAMB R11,ALTCYL              ;AND THE ALTERNATE CYLINDER
00132 FFFFF E0139   859             JMP SEEK03
                    860   ;
00133 373FB F0012   861             RDRAMB R11,SEN409              ;GET DESIRED CYLINDER LSB'S
00134 373F0 F0011   862             RDRAMB R0,SEN408               ;GET DESIRED CYLINDER MSB'S
00135 01700 4F002   863             ADD AB,LSBZ,R0,R0 & PSHLD 2    ;MOVE THE MSB'S...
```

Addr Line CDC O.S. - SEEK ROUTINE

```
                    864             LSHFT2 R0,R0                    ;INTO POSITION
00136 01700 887FF   864 +           ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
00137 3130B FFFFF   865             OR AB,B,R0,R11                  ;MERGE WITH THE LSB'S
00138 373F1 F0014   866             RDRAMB R1,SEN411                ;GET THE PRIMARY HEAD
                    867     ;
                    868     SEEK03: JSR.P CHK2NDA                   ;CHECK FOR SECOND DRIVE
00139 E0403 1F192   868 +           SETPGE CHK2NDA_SHR_H#B & JSRP CHK2NDA
                    869     ;
                    870     ;   SET HEAD
                    871     ;
0013A 45050 F0C00   872             TSTRBIT R0,H#C00                ;ISOLATE CYL. 2 MSB'S IN Q
0013B 325F2 3313F   873             OR ZQ,RSBZ,,R2 & BOZ $+4        ;IF NEITHER, JUMP
0013C 335F2 FFFFF   874             OR ZB,RSBZ,,R2                  ;MOVE MSB'S...
0013D 335F2 FFFFF   875             OR ZB,RSBZ,,R2                  ;INTO POSITION AND...
0013E 31321 FFFFF   876             OR AB,B,R2,R1                   ;MERGE WITH THE HEAD
0013F 9221F 4F004   877             WRTL HSSMA,LDDRB0,R1 & PSHLD 4  ;LOAD THE HEAD NO. AND...
00140 E1009 887FF   878             CTL HSSM,SETAG2 & CLPCNT EVCNTZ ;SET HEAD
00141 45351 F001F   879             MSKRBIT R1,1F                   ;ISOLATE HEAD ONLY
00142 E1013 FFFFF   880             CTL HSSM,RSTAG2                 ;Reset Tag 2
                    881     ;
                    882             JSR.P   RATSEEK                 ;Caching code
00143 E0403 1F107   882 +           SETPGE RATSEEK_SHR_H#B & JSRP RATSEEK
                    883     ;
00144 4505E F0200   884             TSTRBIT R14,200                 ;ALT. TRACK FLAG?
00145 FFFFF 33949   885             BNZ SEEK04                      ;If set, jump
00146 FFFFF 1F1F1   886             JSR HEADRD                      ;GET THE LAST SAVED HEAD...
00147 6002F FFFFF   887             XOR AQ,Q,R2                     ;AND DELETE IT
00148 3011D FFFFF   888             OR AQ,FOUTL,R1,RAMI             ;SAVE CURRENT HEAD
                    889     ;
                    890     SEEK04: JSR.P CKONCYL                   ;CHECK FOR ON-CYL. & 2ND DRIVE
00149 E0404 1F038   890 +           SETPGE CKONCYL_SHR_H#B & JSRP CKONCYL
                    891     ;
                    892     ;   Start the Seek
                    893     ;
0014A 45350 F03FF   894             MSKRBIT  R0,3FF                 ;ISOLATE 10 LSB'S OF CYLINDER
0014B 9220F 4F006   895             WRTL HSSMA,LDDRB0,R0 & PSHLD 6  ;LOAD CYLINDER NO.
0014C E1008 887FF   896             CTL HSSM,SETAG1 & CLPCNT EVCNTZ ;START SEEK
0014D 4505F F0008   897             TSTRBIT R15,8                   ;IF RTZ FLAG SET...
0014E 603FF 33153   898             XOR AQ,B,R15,R15 & BOZ SEEK04A  ;RESET IT & IF NOT - CONTINUE
0014F E1012 1F1F5   899             CTL HSSM,RSTAG1 & JSR CYLSAV    ;SAVE CURRENT CYLINDER
00150 4505F F2000   900             TSTRBIT R15,2000                ;POLLING FLAG SET?
                    901             BNZ.P POLL14B                   ;YES, RETURN TO POLLING
00151 E0401 33BDC   901 +           SETPGE POLL14B_SHR_H#B & BNZP POLL14B
00152 FFFFF E0188   902             JMP SEEK30                      ;ELSE, WAIT FOR THE SEEK
                    903     ;
00153 E1012 4F009   904     SEEK04A: CTL HSSM,RSTAG1 & PSHLD 9      ;RESET TAG 1
00154 C1041 887FF   905             RD HSSM,RDDRST1,R1 & CLPCNT EVCNTZ ;2 µsec. DELAY FOR ON-CYL. TO DROP
                    906     ;
00155 FFFFF C0064   907             LDCT 100                        ;SET 100-µsec. TIMER
00156 3411F F000E   908             ALDRAM R1,SEN114                ;SAVE DRIVE STATUS FOR SENSE
```

Addr Line CDC O.S. - SEEK ROUTINE

```
00157 45051 F000E  909            TSTRBIT R1,ONCYL+SEEKERR+FAULT    ;ISOLATE ON CYL.,SEEK ERR.& FAULT
00158 6605F F0002  910            TSTQMSK ONCYL                     ;CHECK ON CYL. ONLY
00159 C1041 3317F  911            RD HSSM,RDDRST1,R1 & BOZ SEEK08   ;IF ON CYL. ONLY, JUMP
0015A C1041 98156  912            RD HSSM,RDDRST1,R1 & CJPCNT EVCNTZ,$-4 ;DRIVE STATUS & DECR. TIMER
                   913   ;
0015B 45051 F000C  914            TSTRBIT R1,SEEKERR+FAULT          ;CHECK FOR SEEK ERROR OR FAULT
0015C 37160 33992  915            MOVE SMCST,ZERO & BNZ SEEK31      ;IF EITHER, JUMP
                   916   ;
0015D FFFFF 1F1C1  917   SEEK04B:  JSR SEEKLOG                       ;INCR. SEEK LOG COUNT
                   918   ;
0015E 4505F F0080  919   SEEK05:   TSTRBIT R15,80                    ;IF SEEK ONLY FLAG...
0015F 433F2 33188  920            RESETR R2 & BOZ SEEK30            ;NOT SET, WAIT FOR THE SEEK
                   921   ;
                   922   SEEK06:   JSR.P GETDRBIT                    ;GET THE BIT FOR THIS DRIVE
00160 E0401 1F348  922 +          SETPGE GETDRBIT_SHR_H#B & JSRP GETDRBIT
00161 373F1 F00BA  923            RDRAMB R1,ILDFLG                  ;GET ILD FLAGS
00162 45151 F0008  924            TSTBIT R1,8                       ;CHECK FOR ILD SEEK INITIATE
00163 FFFFF 33169  925            BOZ SEEK06B                       ;IF NOT, JUMP
00164 35351 F0002  926            SETRBIT R1,2                      ;SET ILD SEEK COMP. PEND. FLAG
00165 3411F F00BA  927            ALDRAM R1,ILDFLG
                   928   ;
00166 3714E F0063  929            MOVE MSK1,IREG & NOP SKILD        ;SET ILD SEEK COMPLETE...
00167 FFFFF 1F044  930            JSR SMBWRT                        ;PENDING BIT FOR THIS DRIVE
00168 FFFFF E0174  931            JMP SEEK07
                   932   ;
00169 45151 F0004  933   SEEK06B:  TSTBIT R1,4                       ;CHECK ILD RUNNING
0016A FFFFF 33974  934            BNZ SEEK07                        ;IF IT IS, JUMP
                   935            JSR.P SECSAV+1                    ;SAVE SECTOR NO. THIS DRIVE
0016B E0403 1F37F  935 +          SETPGE SECSAV+1_SHR_H#B & JSRP SECSAV+1
                   936   ;
                   937   ;   Set up Seek Complete Pending
                   938   ;
0016C 37352 F0054  939   SEEK06C:  RDMSKB R2,SKPEND                  ;ADD THE BIT FOR THIS DRIVE...
0016D FFFFF 1F1FF  940            JSR ADDABIT                       ;TO SEEK COMP. PENDING
                   941   ;
0016E 37352 F0057  942            RDMSKB R2,SKHIO1
0016F 375F1 4002E  943            OR DZ,RSBZ,RAME,R1 & PUSH BITHIO  ;GET BIT FOR THIS HAIO
00170 335F1 FFFFF  944            OR ZB,RSBZ,,R1                    ;COMPUTE THE BIAS...
00171 FFFFF B3173  945            CJPOP ALUZ,$+2                    ;FOR SEEK COMPLETE...
00172 05342 D0003  946            ADD DB,B,MSK1,R2 & LOOP H#3       ;HAIO BITS
00173 FFFFF 1F1FF  947            JSR ADDABIT                       ;ADD SEEK COMP. THIS DR., THIS HAIO
                   948   ;
00174 5535F F0080  949   SEEK07:   RSTRBIT R15,80                    ;RESET SEEK ONLY FLAG
00175 FFFFF 1F1F5  950            JSR CYLSAV                        ;SAVE CURRENT CYL. THIS DRIVE
                   954   ;
00176 E100B FFFFF  955            CTL HSSM,RSTUSEL                  ;DE-SELECT THE DRIVE
00177 3710F F0302  956            RESETRAM SEEKONL                  ;Clear Seek Only flag
                   957   ;
00178 3714E F008D  958            MOVE MSK1,IREG & NOP SKERRBIT     ;GET SEEK ERROR...
00179 FFFFF 1F03B  959            JSR SMBRD                         ;BITS FOR ALL DRIVES
0017A 41340 3317D  960            AND AB,B,R4,R0 & BOZ $+3          ;IF NONE, JUMP
```

```
 Addr            Line CDC O.S. - SEEK ROUTINE

0017B 40331 3381A   961          AND AQ,B,R3,R1 & BNZ WAIT14      ;IF SEEK ERROR THIS DRIVE...
0017C FFFFF 3381A   962          BNZ WAIT14                       ;RETURN TO WAIT LOOP
                    963    ;
0017D 3710F F002C   964          RESETRAM SKRETRY                 ;RESET RETRY COUNT
0017E 433F5 E0D68   965          RESETR   R5 & JMP END20          ;Send Chan. End
```

Addr Line CDC O.S. - SEEK ROUTINE

```
                    967   ;
0017F 4505F F0080   968   SEEK08:   TSTRBIT R15,80                  ;CHECK FOR SEEK ONLY FLAG
00180 FFFFF 3318B   969             BOZ SEEK30                      ;IF NOT SET, JUMP
00181 FFFFF 1F1F5   970             JSR CYLSAV                      ;SAVE CURRENT CYLINDER
00182 3735A F0005   971   SEEK08A:  RDMSKB R10,5                    ;STATUS ACTION = 5
                    981   ;
                    982   ;   Send Channel End / Device End
                    983   ;
00183 3535D F0800   984   SEEK10:   SETRBIT R13,800                 ;SET STATUS = CHAN. END
00184 FFFFF E0D75   985             JMP END40A                      ;ADD DEVICE END & SEND STATUS
                    986
                    987
00185 3535F F0080   988   SEEK20:   SETRBIT R15,80                  ;SET SEEK ONLY FLAG
                    994   ;
                    995   ;   Rezero (RTZ) the drive
                    996   ;
00186 3535F F0008   997   SEEK22:   SETRBIT R15,8                   ;SET RTZ FLAG
00187 B2200 F0050  1002             WRTE HSSMA,LDDRB0 & NOP H#50    ;BUS TO RE-ZERO & FAULT RESET
00188 FFFFF 4F006  1003             PSHLD 6
00189 E100A 887FF  1004             CTL HSSM,SETAG3 & CLPCNT EVCNTZ ;RE-ZERO COMMAND
0018A E1016 E015E  1005             CTL HSSM,RSTAG3 & JMP SEEK05
                   1006
                   1007
0018B C1043 FFFFF  1008   SEEK30:   RD HSSM,RDDRST1,R3              ;GET DRIVE STATUS
0018C C1043 FFFFF  1009             RD HSSM,RDDRST1,R3
0018D 45053 F0020  1010             TSTRBIT R3,SELECT               ;CHECK UNIT SELECTED
0018E FFFFF 33992  1011             BNZ SEEK31                      ;IF SELECTED, CONTINUE
                   1012   ;
0018F 3710F F0008  1013   SEEK30A:  RESETRAM SEN108                 ;CLEAR SENSE 8, FORMAT 1
00190 3413F F000E  1014             ALDRAM R3,SEN114                ;SAVE DRIVE STATUS FOR SENSE
                   1015             JPOP.P SEL35                    ;EQUIP. CHECK
00191 E0401 BF327  1015 +           SETPGE SEL35_SHR_H#B & JPOPP SEL35
                   1016   ;
00192 373F1 F002C  1017   SEEK31:   RDRAMB R1,SKRETRY               ;GET SEEK RETRY COUNT
00193 37352 F0041  1018             RDMSKB R2,READY+BUSY            ;SET 'READY & BUSY' MASK
00194 3705F F003D  1019             RDMSK@ 3D                       ;SET UP...
00195 D0FF0 5F197  1020             RDXFR SMC,,R0 & LDAR $+2        ;500MS. TIMER
00196 35350 F0090  1021             SETRBIT R0,090
00197 C1043 331D5  1022             RD HSSM,RDDRST1,R3 & BOZ SEEK38 ;IF TIME-OUT, JUMP
00198 C1043 C0009  1023             RD HSSM,RDDRST1,R3 & LDCT 9     ;READ DRIVE STATUS
00199 45053 F0004  1024             TSTRBIT R3,SEEKERR              ;SEEK ERROR?
0019A 333F1 339CB  1025             OR ZB,B,,R1 & BNZ SEEK36        ;YES, PROCESS IT
0019B 45053 F0008  1026             TSTRBIT R3,FAULT                ;DRIVE FAULT?
0019C 41032 139B7  1027             AND AB,Q,R3,R2 & CJSR NALUZ,SEEK35 ;YES, ATTEMPT TO RESET IT
0019D 37160 339A1  1028             MOVE SMCST,ZERO & BNZ SEEK31A   ;IF READY OR BUSY, CONTINUE
                   1029   ;
0019E C1043 9819E  1030             RD HSSM,RDDRST1,R3 & CJPCNT EVCNTZ,$ ;DELAY FOR GOOD STATUS
```

Addr Line CDC O.S. - SEEK ROUTINE

```
0019F 41032 FFFFF  1031            AND AB,Q,R3,R2                    ;CHECK FOR READY OR BUSY
001A0 37160 3318F  1032            MOVE SMCST,ZERO & BOZ SEEK30A     ;IF NEITHER, EQUIP. CHECK
                   1033    ;
001A1 45053 F0002  1034    SEEK31A: TSTRBIT R3,ONCYL                  ;CHECK ON CYLINDER
001A2 133F0 339A4  1035            DECRB R0 & BNZ $+2                ;IF ON CYL., JUMP; DECR. TIMER
001A3 333F0 731D5  1036            OR ZB,B,,R0 & CJARB ALUZ,SEEK38   ;IF TIME OUT, ERROR (SEEK38)
                   1037    ;
001A4 45053 F0080  1038            TSTRBIT R3,SKEND                  ;SEEK END ALSO?
001A5 333F0 739A6  1039            OR ZB,B,,R0 & CJARB NALUZ,$+1     ;YES, CONTINUE ($+1)
001A6 FFFFF 1F1F5  1040            JSR CYLSAV                        ;SAVE CUR. CYLINDER THIS DRIVE
001A7 4505F F0008  1041            TSTRBIT R15,8                     ;CHECK RTZ FLAG
001A8 FFFFF 339B0  1042            BNZ SEEK32                        ;IF SET, RE-SEEK THE DRIVE
001A9 3413F F000E  1043            ALDRAM R3,SEN114                  ;NORMAL STATUS FOR SENSE
                   1044            ADDMBIT 10,SEN108                 ;ADD SEEK COMPLETE to Sense 8
001AA 373F0 F0008  1044 +          RDRAMB   R0,SEN108
001AB 35350 F0010  1044 +          SETRBIT  R0,10
001AC 3410F F0008  1044 +          ALDRAM   R0,SEN108
001AD 4505F F0080  1045            TSTRBIT R15,80                    ;CHECK SEEK ONLY FLAG
001AE 3710F A302C  1046            MOVE ZERO,RAME & CRTS ALUZ,SKRETRY ;IF NOT, RETURN; CLR. RETRY COUNT
001AF FFFFF E017F  1051            JMP SEEK08                        ;NORMAL EXIT
                   1052    ;
                   1053    SEEK32:  EQU $
001B0 3710D FFFFF  1057            MOVE ZERO,RAMI                    ;RESET CYL. BEFORE RE-SEEKING
001B1 4505E F0200  1058            TSTRBIT  R14,200                  ;Alternate track flag?
001B2 FFFFF 33985  1059            BNZ      $+3                      ;Yes, jump
001B3 341BF F0012  1060            ALDRAM R11,SEN409                 ;SAVE CUR. CYL. FOR THE SEEK
001B4 3710F F0011  1061            RESETRAM SEN408                   ;RESET CYL. MSB'S
001B5 0F10F F0302  1062            ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag
001B6 FFFFF E012A  1063            JMP SEEK02                        ;AND RE-SEEK THE DRIVE
                   1064    ;
                   1065    ;
001B7 B2200 F0010  1066    SEEK35:  WRTE HSSMA,LDDRB0 & NOP H#10      ;BUS OUT FOR FAULT RESET
001B8 E100A 4F009  1067            CTL HSSM,SETAG3 & PSHLD 9         ;SET TAG TO RESET FAULT
001B9 FFFFF 887FF  1068            CLPCNT   EVCNTZ                   ;Delay while it works
001BA E1016 C0004  1069            CTL      HSSM,RSTAG3 & LDCT 4
001BB FFFFF 399BF  1070            CJMP     LOG,SEEK35A              ;If Forced Logging, error
001BC C1043 981BC  1071            RD HSSM,RDDRST1,R3 & CJPCNT EVCNTZ,$ ;Wait for stable status
001BD 45153 F0008  1072            TSTBIT R3,FAULT                   ;CHECK FOR FAULT
001BE 45043 A3001  1073            AND DB,Q,MSK1,R3 & CRTS ALUZ,READY ;IF FAULT CLEARED, RETURN
001BF 34335 8F7FF  1074    SEEK35A: RTOR R3,R5 & POP                  ;STATUS TO R5 & POP THE STACK
                   1075            JPOP.P SEL31                      ;EQUIP. CHECK/DEVICE CHECK
001C0 E0401 BF310  1075 +          SETPGE SEL31_SHR_H#B & JPOPP SEL31
                   1076
001C1 FFFFF 1F350  1077    SEEKLOG: JSR ERRLOG                        ;LOC. OF ERROR/USAGE LOG
001C2 0614E F0007  1078            ADD DQ,FOUTL,MSK1,IREG & NOP D#7  ;LOC. OF SEEK COUNT LSB'S
001C3 3705F F0010  1079            RDMSKQ 10                         ;SEEK COUNT INCREMENT
001C4 060DF FFFFF  1080            ADD DQ,Q,RAMI                     ;INCR. THE SEEK COUNT
001C5 321FD A77FF  1081            QLDREG RAMI & CRTS UCZ            ;IF LSB'S DON'T OVERFLOW, RETURN
001C6 271EE FFFFF  1082            DECRD IREG,IREG                   ;LOC. OF SEEK COUNT MSB'S
```

```
 Addr            Line CDC O.S. - SEEK ROUTINE

001C7 0F0DF FFFFF  1083            ADD DZC,0,RAMI                  ;INCR. THE MSB'S
001C8 46150 F0080  1084            TESTQ 80                        ;SEEK COUNT = 8000-HEX?
001C9 321FD A37FF  1085            QLDREG RAMI & CRTS ALUZ         ;IF NOT, RETURN
001CA FFFFF E2025  1086            JMP LOGOFL                      ;ELSE, SET LOG OVERFLOW THIS DRIVE
                   1087
                   1088    ; SEEK ERROR PROCESSING
                   1089    ;
001CB FFFFF 131EB  1090    SEEK36: CJSR ALUZ,SKERLOG               ;IF NO RETRIES, UPDATE ERROR LOG
                   1091    SEEK37: CJMP.P  LOG,SEL32               ;If Forced logging, no retries
001CC E0401 39B19  1091 +          SETPGE SEL32_SHR_H#B & CJMPP LOG,SEL32
001CD 373F1 F002C  1092            RDRAMB R1,SKRETRY               ;GET SEEK RETRY COUNT
001CE 45051 F0002  1093            TSTRBIT R1,2                    ;CHECK FOR 2 RETRIES
001CF FFFFF B39D2  1094            CJPOP NALUZ,$+3                 ;IF 2, PERMANENT ERROR
001D0 0C11F F002C  1095            ADD AZC,FOUTL,R1,RAME & NOP SKRETRY ;INCR. RETRY COUNT
001D1 FFFFF E0186  1096            JMP SEEK22                      ;RE-ZERO THE DRIVE
                   1097    ;
                   1098            SETMBIT 80,SEN01                ;SET SENSE 1 = Permanent Error
001D2 37350 F0080  1098 +          RDMSKB  R0,80
001D3 3410F F0001  1098 +          ALDRAM  R0,SEN01
001D4 FFFFF E0B19  1099            JMP SEL32
                   1100    ;
001D5 4505F F0008  1101    SEEK38: TSTRBIT R15,8                   ;RTZ FLAG SET?
001D6 FFFFF 331CC  1102            BOZ SEEK37                      ;IF NOT, RE-ZERO THE DRIVE
001D7 373F0 F0008  1103            RDRAMB R0,SEN108                ;GET SENSE 8, FORMAT 1
001D8 35350 F0020  1104            SETRBIT R0,20                   ;SET SEEK INCOMPLETE
001D9 FFFFF E0B1A  1105            JMP SEL32+1
```

Addr Line CDC O.S. -- SEEK ROUTINE

```
                    1107    ;    THIS SUBROUTINE SETS UP THE SENSE BYTES FOR DOING A SEEK.
                    1108    ;UPON ENTRY:  R11 = CYLINDER
                    1109    ;             R12 = HEAD
                    1111    ;< REGISTERS USED:  R0,R2,Q & IREG.
                    1115    ;UPON EXIT:  R0-UPPER = CYLINDER MSB'S
                    1116    ;            Q  = CYLINDER 8 LSB'S
                    1117    ;
001DA 4505B F00FF   1118    SEEKSENS: TSTRBIT R11,H#FF                  ;ISOLATE CYL. LSB'S IN Q AND...
001DB 321FF F0012   1119              QLDRAM SEN409                     ;SAVE THEM FOR SENSE 9, FORMAT 4,5
001DC 4515E F0200   1121              TSTBIT R14,200                    ;<DEFECTIVE TRACK FLAG? (TO IREG)
001DD FFFFF 339E0   1122              BNZ $+3                           ;<YES, JUMP
001DE 321FF F0005   1123              QLDRAM SEN05                      ;<SAVE CYL. LSB'S FOR SENSE 5
001DF 605B2 FFFFF   1124              XOR AQ,RSBZ,R11,R2                ;<CYL. MSB'S INTO R2 (SHIFTED)
001E0 603B0 4F001   1126              XOR AQ,B,R11,R0 & PSHLD 1         ;ISOLATE CYL. MSB'S IN R0 AND...
                    1127              LSHFT2 R0                         ;MOVE THEM INTO POSITION
001E1 01700 887FF   1127 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
001E2 3490F F0011   1128              ALDRAMU R0,SEN408                 ;SAVE THEM FOR SENSE 8, FORMAT 4,5
001E3 341CF F0014   1130              ALDRAM R12,SEN411                 ;<SAVE HEAD FOR SENSE
001E4 371EE FFFFF   1131              MOVE IREG,IREG                    ;<DEFECTIVE TRACK FLAG?
001E5 335F2 A3FFF   1132              OR ZB,RSBZ,,R2 & CRTS NALUZ       ;<YES, RETURN; MOVE CYL. MSB'S...
001E6 335F2 FFFFF   1133              OR ZB,RSBZ,,R2                    ;<INTO POSITION AND...
001E7 313C2 FFFFF   1134              OR AB,B,R12,R2                    ;<MERGE WITH HEAD NO. INTO R2
001E8 3412F AF006   1135              ALDREG R2,RAME & RTS SEN06        ;<SAVE FOR SENSE & RETURN
                    1139
001E9 FFFFF 1F1F9   1140    CYL00:    JSR LOGICAL                       ;GET LOGICAL DRIVE NO.
001EA 0614E AF030   1141              ADD DQ,FOUTL,MSK1,IREG & RTS H#30 ;COMPUTE LOC. OF CUR. CYL.
                    1142
                    1143    ; SEEK ERROR LOGGING
                    1144
001EB FFFFF 1F350   1145    SKERLOG:  JSR ERRLOG                        ;GET ERROR/USAGE LOC. THIS DRIVE
001EC 0614E F0008   1146              ADD DQ,FOUTL,MSK1,IREG & NOP H#8  ;MODIFY ADDR. FOR SEEK LOG
001ED 0F3D0 FFFFF   1147              ADD DZC,B,RAMI,R0                 ;INCR. SEEK ERROR COUNT
001EE 45050 F0008   1148              TSTRBIT R0,8                      ;CHECK FOR 8 SEEK ERRORS
001EF 3410D A37FF   1149              ALDREG R0,RAMI & CRTS ALUZ        ;SAVE NEW COUNT; IF <8, RETURN
001F0 FFFFF E2025   1150              JMP LOGOFL                        ;SET LOG OVERFLOW THIS DRIVE
                    1151
001F1 FFFFF 1F1F9   1152    HEADRD:   JSR LOGICAL
001F2 0614E F03A2   1153              ADD DQ,FOUTL,MSK1,IREG & NOP UNITHD ;UNIT/HEAD INDEXED BY DRIVE NO.
001F3 370DF FFFFF   1154              RDREGQ RAMI                       ;READ UNIT NO. & HEAD NO.
001F4 46342 AF01F   1155              AND DQ,B,MSK1,R2 & RTS H#1F       ;ISOLATE HEAD NO. IN R2 & RETURN
                    1156    ;
001F5 4505E F0200   1157    CYLSAV:   TSTRBIT R14,200                   ;CHECK ALT. TRACK FLAG
001F6 FFFFF A3FFF   1158              CRTS NALUZ                        ;IF SET, DON'T SAVE CYLINDER
001F7 FFFFF 1F1E9   1159              JSR CYL00                         ;LOC. OF CUR. CYL. THIS DRIVE
001F8 341BD AF7FF   1160              ALDREG R11,RAMI & RTS             ;SAVE CURRENT CYLINDER & RETURN
```

```
 Addr            Line CDC O.S. - SEEK ROUTINE


                 1162   ;    THIS ROUTINE RETURNS WITH THE LOGICAL (NOT PHYSICAL) DRIVE NO. IN Q.
                 1163   ;IT WILL NOT ALTER THE DRIVE NO. IN R13.  USES IREG.
                 1164   ;
001F9 4505D F0880 1165   LOGICAL:  TSTRBIT R13,VALID+H#80            ;VALID BIT OR S.S.S. UNIT?
001FA 4504D A387F 1166            AND DB,Q,MSK1,R13 & CRTS NALUZ,H#7F ;YES, RETURN; ISOLATE DRIVE
001FB 371F0 F02F4 1167            MOVE RAME,ZERO & NOP TWODRIV       ;TWO DRIVES PER UNIT?
001FC 3410E A37FF 1168            ALDREG R0,IREG & CRTS ALUZ         ;IF NOT,RETURN
001FD 324F0 FFFFF 1169            OR ZQ,RSQBZ,,R0                    ;1/2 PHYSICAL UNIT = LOGICAL DRIVE
001FE 373E0 AF7FF 1170            RDREGB IREG,R0 & RTS               ;RETURN
                 1171
                 1172
                 1173   ;    THIS ROUTINE WILL ADD A BIT FOR THIS DRIVE INTO A WORD IN SMB MEMORY.
                 1174   ;
                 1175   ;UPON ENTRY:  R2 = 1ST SMB LOCATION
                 1176   ;             R3 = BIT FOR THIS DRIVE (24-31)
                 1177   ;             R4 = BIT FOR THIS DRIVE (0-23)
                 1178   ;REGISTERS USED:  R0,R1,Q & IREG.
                 1179   ;
001FF 3412E FFFFF 1180   ADDABIT:  ALDREG R2,IREG                    ;SMB LOC. TO IREG
00200 37160 1F03B 1181            MOVE SMCST,ZERO & JSR SMBRD        ;GET THE WORD AND...
00201 31340 FFFFF 1182            OR AB,B,R4,R0                      ;ADD THE BIT FOR THIS DRIVE (0-23)
00202 3003F FFFFF 1183            OR AQ,Q,R3                         ;ADD THE BIT FOR THIS DRIVE (24-31)
00203 3412E E0044 1184            ALDREG R2,IREG & JMP SMBWRT        ;SAVE ALL THE BITS & RETURN
                 1185
                 1186
                 1188   TITLE CDC O.S. - ADDRESS CONVERSION
```

Addr Line CDC O.S. - ADDRESS CONVERSION

```
1197
1198
1199    ;     THE CONVERSION ROUTINE WILL REQUEST AN ADDRESS WORD FROM THE CHANNEL.
1200    ;THE ADDRESS WORD WILL BE IN R0 (BITS 0-23) & Q (BITS 24-35).
1201    ;THE WORD WILL BE CHECKED FOR ILLEGAL VALUES.  IF ILLEGAL, STATUS OF
1202    ;"COMMAND REJECT", "DATA VALUE NOT AS REQUIRED" WILL BE GENERATED AND
1203    ;IMMEDIATELY PRESENTED TO THE CHANNEL.
1204    ;
1205    ;     IF NO ADDRESS WORD IS RECEIVED WITHIN 2 MSEC. FROM WHEN ONE IS REQUESTED
1206    ;(ODR SET), "COMMAND REJECT" STATUS WITH "CHAN. TRUNCATION" (BIT 31) WILL
1207    ;BE PRESENTED.
1208    ;
1210    ;    ROUTINE CONV05 (WITH CONV20-26) WILL CONVERT A RELATIVE ADDRESS TO
1211    ;CYLINDER, HEAD, RECORD & WORD NUMBER.  THE CYLINDER WILL BE SAVED
1219    ;IN SENSE BYTES 8 & 9, FORMAT 4,5.  THE HEAD WILL BE SAVED IN SENSE BYTE 11,
1220    ;FORMAT 4,5.  THE RECORD NO. WILL BE IN SENSE 12, FORMAT 4,5.  RECORDS ARE
1221    ;NUMBERED 1-MAX.; THERE IS NO RECORD 0.  THE WORD NO. WILL BE IN R6.
1222    ;
1223    ;     ROUTINES CONV10-15 WILL CHECK ABSOLUTE ADDRESSING VALUES & CONVERT
1224    ;THESE VALUES TO A RELATIVE WORD NO.  THE ABSOLUTE CYLINDER & HEAD
1225    ;WILL BE SAVED IN SENSE BYTES 5 & 6 WHETHER THEY ARE VALID OR NOT.
1226    ;     THE RELATIVE WORD NO. WILL BE CONVERTED (USING CONV05) TO AMPERIF CYL.,
1227    ;HEAD, RECORD & WORD NO. (ALWAYS 0) AS SHOWN ABOVE.
1228
1229
1230    ;    REGISTERS:
1231    ;         Q = DIFFERENCE REG.
1232    ;        R0 = DIVIDEND REG. (10 LSB'S)
1233    ;        R1 = DIVIDEND REG. (17 MSB'S)
1234    ;        R2 = DIVISOR REG.
1235    ;        R3 = RECORD NO. OR SECTOR NO. (USUALLY 0)
1236    ;        R4 = HEAD NO.
1237    ;        R5 = ITERATION COUNT
1238    ;        R6 = STARTING WORD NO. (AT END OF ROUTINE)
1239    ;        R7 = TRANSFER LENGTH (FROM FUNCT. WORD)
1240    ;        R8 = CYLINDER DIVISOR
1241    ;        R9 = HEAD DIVISOR
1242    ;        R9 = RECORD NO. (END OF ROUTINE)
1243    ;       R10 = SCRATCH
1244    ;       R11 = CYLINDER NO.
1245    ;       R12 = COMMAND CODE (BEGINNING OF ROUTINE)
1246    ;       R12 = AMPERIF HEAD (END OF ROUTINE)
1247    ;       R13 = DRIVE NO.
1248    ;      R14,R15 = FLAG REGISTERS
1249    ;       IREG = SCRATCH
1250
```

Addr Line CDC O.S. - ADDRESS CONVERSION

```
                1252
                1253
00204 3535F F0004 1254  CONV00:  SETRBIT R15,4                    ;SET ABSOLUTE ADDRESS FLAG
00205 37385 FFFFF 1255  CONV01:  RDREGB HIOST,R5                  ;GET HAIO STATUS
00206 45055 F0040 1256           TSTRBIT R5,40                    ;ADDRESS WORD AVAILABLE?
00207 FFFFF 33A0F 1257           BNZ CONV02                       ;IF YES, CONTINUE
                1258  ;
00208 E417E 4F7FF 1259           CTL HIO,SETODR & PSHLD H#7FF     ;SET ODR & A TIMER
00209 FFFFF FFFFF 1260           NOP
0020A FFFFF FFFFF 1261           NOP
0020B 45055 F1C44 1262           TSTRBIT R5,1C44                  ;ALLOW ONLY SPECIFIED BITS
0020C 37385 B3A0F 1263           RDREGB HIOST,R5 & CJPOP NALUZ,CONV02 ;GET HAIO STATUS; IF ANY, JUMP
0020D FFFFF 887FF 1264           CLPCNT EVCNTZ                    ;IF NO TIME-OUT, CONTINUE WAITING
                1265  ;
                1266  ;  2 M-SEC. TIME-OUT
                1267  ;
0020E E417D E02A8 1268           CTL HIO,RSTODID & JMP CONV30     ;RESET ODR; NO ADDRESS RCVD.
                1269  ;
0020F 45055 F1C80 1270  CONV02:  TSTRBIT R5,1C80                  ;CHECK FOR BAD INTERRUPTS
00210 E417D 33AA8 1271           CTL HIO,RSTODID & BNZ CONV30     ;IF ANY, ERROR; RESET ODR
                1272  ;
00211 E4404 FFFFF 1273           CTL HIO,HIOTGTS                  ;REQUEST E.F. ADDRESS WORD
00212 E4208 FFFFF 1274           CTL HIO,EFREQS                   ;REQUEST E.F. ADDRESS WORD
00213 37351 F0020 1275           RDMSKB R1,20                     ;MASK FOR E.F. AVAIL.
00214 45081 4F006 1276           AND DB,Q,HIOST,R1 & PSHLD 6      ;CHECK E.F. AVAIL.
00215 3706F B3A18 1277           RDREGQ SMCST &  CJPOP NALUZ,$+3  ;IF E.F. AVAIL., JUMP
00216 45081 887FF 1278           AND DB,Q,HIOST,R1 & CLPCNT EVCNTZ ;IF *TIME-OUT, CONTINUE WAITING
                1279
00217 FFFFF E006A 1280           JMP EFPROCA                      ;TIME-OUT; EQUIP. CHECK
                1281
00218 37352 F0288 1282           RDMSKB R2,EFADDRES               ;LOC. OF ADDRESS WORDS
00219 373A4 FFFFF 1283           RDREGB UBR1,R4                   ;GET ADDR. 24 LSB'S
0021A 373B3 FFFFF 1284           RDREGB UBR2,R3                   ;GET ADDR. 12 MSB'S
0021B 37381 FFFFF 1285           RDREGB HIOST,R1                  ;GET HIO STATUS
0021C 45151 F0001 1286           TSTBIT R1,1                      ;CHECK FOR CHANNEL PARITY
0021D E4200 3390B 1287           CTL HIO,EFREQR & BNZ PARERR1     ;IF PARITY ERROR, UNIT CHECK
                1288  ;
0021E 37361 FFFFF 1289           RDREGB SMCST,R1                  ;READ XMC STATUS
0021F 45351 F000B 1290           MSKRBIT R1,H#B                   ;CHECK PARITY ERRORS
00220 FFFFF 33222 1291           BOZ $+2                          ;IF NONE, CONTINUE
00221 34313 E03BA 1292           RTOR R1,R3 & JMP IRPT12+1        ;PROCESS PARITY ERROR
                1293  ;
00222 45053 F0800 1294           TSTRBIT R3,800                   ;CHECK FOR BIT 35
00223 E4408 33AAD 1295           CTL HIO,HIOTGTR & BNZ CONV37     ;IF BIT 35, FORCED E.F.
                1296  ;
00224 3412E 1F03B 1310  CONV02A: ALDREG R2,IREG & JSR SMBRD        ;GET PREVIOUS ADDR. WORD
00225 61140 FFFFF 1311           XOR AB,FOUTL,R4,R0               ;24 LSB'S THE SAME?
00226 34340 33A29 1312           RTOR R4,R0 & BNZ $+3             ;IF NOT, JUMP
00227 60130 FFFFF 1313           XOR AQ,FOUTL,R3,ZERO             ;12 MSB'S THE SAME?
00228 34331 132B1 1314           RTOR R3,R1 & CJSR ALUZ,CONV40    ;YES, JUMP
00229 3412E 1F356 1315           ALDREG R2,IREG & JSR AGE00       ;AGE ADDR. WORD DISPLAYS
                1316  ;
```

```
 Addr            Line CDC O.S. - ADDRESS CONVERSION

0022A 323F1 FFFFF  1317            QTOR R1                          ;MOVE Q-REG. TO R1
0022B 4505D F003F  1323            TSTRBIT R13,3F                   ;ISOLATE DRIVE NO. (PHYSICAL)...
0022C 321FF F0205  1324            QLDRAM LASTDRV                   ;AND SAVE IT
0022D 4505F F0004  1325            TSTRBIT R15,4                    ;CHECK ABSOLUTE ADDRESS FLAG
0022E FFFFF 33A52  1326            BNZ CONV10                       ;IF SET, JUMP
                   1327
                   1328    ;    RELATIVE ADDRESS PROCESSING
                   1329
0022F 2D0F1 F02DF  1339            SUBB DB,Q,RAME,R1 & NOP LIMIT1   ;IF ADDRESS MUCH TOO LARGE...
00230 FFFFF 352AB  1340            CJMP ALB,CONV35                  ;SEND COMMAND REJECT
00231 2D0F1 F02DF  1341            SUBB DB,Q,RAME,R1 & NOP LIMIT1   ;IF ADDR. MSB'S LESS THAN MAX...
00232 FFFFF 34238  1343            CJMP AGB,CONV05                  ;<ADDRESS IS VALID
                   1347    ;
00233 370FF F02E0  1348            RDRAMQ LIMIT2                    ;SET UP MAX...
00234 D0FFA FFFFF  1349            RDXFR SMC,,R10                   ;VALID ADDRESS...
00235 353FA F02E1  1350            OR DB,B,RAME,R10 & NOP LIMIT3    ;FOR CHECKING
00236 2930A FFFFF  1351            SUBB AB,B,R0,R10
00237 FFFFF 35AAB  1352            CJMP NALB,CONV35                 ;IF ADDRESS BAD, COMMAND REJECT
                   1382    ;
                   1383    ;    CONVERT AMPERIF CYLINDER, HEAD, RECORD & WORD NO.
                   1384    ;
                   1386    CONV05:  EQU  $                          ;<
                   1391    ;
00238 370FF F02D7  1392    CONV05A: RDRAMQ CYLDIVAM                 ;GET AMPERIF DIVISOR MSB'S
00239 D0FF8 FFFFF  1393            RDXFR SMC,,R8
0023A 357F8 F02D8  1394            OR DB,LSBZ,RAME,R8 & NOP CYLDIVAL ;MERGE LSB'S; MULTIPLY X 2
0023B 337F8 C0002  1395            OR ZB,LSBZ,,R8 & LDCT 2          ;X2 = X4
0023C 373F9 F02D9  1396            RDRAMB R9,WRDSTRK                ;GET 1/4 WORDS PER TRACK...
0023D 01799 9823D  1397            ADD AB,LSBZ,R9,R9 & CJPCNT EVCNTZ,$ ;X64 = AMPERIF HEAD DIVISOR
0023E 3401F 1F288  1398            RTOQ R1 & JSR CONV20             ;SET UP DIVIDEND & DIVISOR REGS.
0023F 343DA C000B  1399            RTOR R13,R10 & LDCT 11           ;Drive no. to R10
00240 FFFFF 1F28D  1400            JSR CONV21                       ;COMPUTE AMPERIF CYL./HD./REC.
                   1401    ;
00241 3434C FFFFF  1402            RTOR R4,R12                      ;HEAD NUMBER TO R12
00242 34314 1F1DA  1403            RTOR R1,R4 & JSR SEEKSENS        ;SAVE CYL./HEAD FOR SENSE
00243 34339 FFFFF  1405            RTOR     R3,R9                   ;$Record no. to R9
00244 321FF F02F8  1410            QLDRAM CSEN409                   ;SAVE INITIAL CYL. 8 LSB'S
00245 3490F F02FA  1411            ALDRAMU R0,CSEN408               ;SAVE INITIAL CYL. MSB'S
```

```
 Addr            Line CDC O.S. - ADDRESS CONVERSION

                 1435
00246 45359 F007F 1436   CONV07:  MSKRBIT  R9,7F                    ;Mask Record no. to 7 bits
00247 0C19F F0015 1437            ADD AZC,FOUTL,R9,RAME & NOP SEN412 ;Record+1 to Sense 12, Format 4,5
00248 0B3F9 FFFFF 1439            INCRB    R9                       ;$Incr. record no.+ 1
00249 3419F F02FD 1444            ALDRAM R9,CSEN412                 ;SAVE INITIAL RECORD NO.
0024A 341CF F02FC 1445            ALDRAM R12,CSEN411                ;Save initial head no.
                 1446   ;
0024B 4505F F0004 1447            TSTRBIT R15,4                     ;CHECK ABSOLUTE ADDRESSING
0024C 603FF 3324E 1448            XOR AQ,B,R15,R15 & BOZ $+2        ;IF NOT ABSOLUTE JUMP
0024D 433F6 E0250 1449            RESETR R6 & JMP $+3               ;SET WORD NO.=0 & JUMP
                 1450   ;
0024E 34546 4F009 1451            OR AZ,RSBZ,R4,R6 & PSHLD 9        ;GET WORD NO. INTO R6 AND...
0024F 335F6 887FF 1452            RSHFTZ R6                         ;MOVE IT INTO POSITION
00250 3416F F02F9 1453            ALDRAM R6,CWORD                   ;SAVE INITIAL WORD
00251 FFFFF E1992 1454            JMP CHK2NDA                       ;CHECK FOR 2ND DRIVE & RETURN
                 1455
                 1456
                 1457   ; ABSOLUTE ADDRESS PROCESSING
                 1458
00252 45050 F00FF 1459   CONV10:  TSTRBIT R0,H#FF                   ;ISOLATE SECT. ADDR./RECORD NO...
00253 323F3 FFFFF 1460            QTOR R3                           ;AND SAVE IT IN R3
00254 6505C F0027 1461            TSTRMSK R12,27                    ;CHECK FOR SEEK & SET SECTOR
00255 FFFFF 33A5A 1462            BNZ CONV11                        ;IF NOT, JUMP
00256 2D043 F0080 1463            SUBB DB,Q,MSK1,R3 & NOP D#128     ;CHECK FOR GOOD VALUE
00257 FFFFF 3425A 1464            CJMP AGB,CONV11                   ;IF VALID SECTOR, CONTINUE
00258 65053 F00FF 1465            TSTRMSK R3,D#255                  ;CHECK FOR SECTOR = 255
00259 FFFFF 33AAB 1466            BNZ CONV35                        ;IF NOT 255, COMMAND REJECT
                 1467   ;
0025A 3401F 4F003 1468   CONV11:  RTOQ R1 & PSHLD 3                 ;MOVE R1 BACK TO Q
0025B 336F0 887FF 1470            LSHFTDZ R0                        ;<MOVE HEAD INTO POSITION
0025C 3490E 4F001 1474            ALDREGU R0,IREG & PSHLD 1         ;SAVE HEAD IN IREG
0025D 336F0 887FF 1476            LSHFTDZ R0                        ;<MOVE CYLINDER INTO POSITION
0025E 4634A F00FF 1480            AND DQ,B,MSK1,R10 & NOP H#FF      ;ISOLATE CYL. 8 LSB'S IN R10...
0025F 341AF F0005 1481            ALDRAM R10,SEN05                  ;AND SAVE THEM FOR SENSE BYTE 5
00260 323FB FFFFF 1486            QTOR R11                          ;SAVE COMPLETE CYLINDER IN R11
00261 600AF 4F002 1487            XOR AQ,Q,R10 & PSHLD 2            ;ISOLATE CYLINDER MSB'S AND...
00262 334F0 887FF 1488            RSHFTDZ R0                        ;MOVE THEM INTO POSITION
00263 373E4 FFFFF 1489            RDREGB IREG,R4
00264 45354 F00FF 1490            MSKRBIT R4,H#FF                   ;ISOLATE HEAD IN R4 FOR CHECKING
00265 3014F F0006 1491            OR AQ,FOUTL,R4,RAME & NOP SEN06   ;MERGE CYL. MSB'S & HEAD TO SENSE 6
```

Addr Line CDC O.S. - ADDRESS CONVERSION

```
                1501    ;
00266 2D0FB F02D3 1503          SUBB DB,Q,RAME,R11 & NOP MAXCYLA   ;<IF CYLINDER NO. GREATER...
00267 FFFFF 352AB 1507          CJMP ALB,CONV35                    ;THAN MAX., COMMAND REJECT
                1508    ;
00268 2D0F4 F02D5 1510          SUBB DB,Q,RAME,R4 & NOP MAXHDA     ;<IF HEAD NO. GREATER...
00269 420FF 352AB 1514          RESETQ & CJMP ALB,CONV35           ;THAN MAX., COMMAND REJECT
                1515    ;
0026A 343B1 FFFFF 1516          RTOR R11,R1                        ;MOVE CYLINDER TO MULTIPLIER
0026B 34318 33271 1517          RTOR R1,R8 & BOZ CONV13            ;IF CYL.= 0, SKIP MULTIPLICATION
0026C 370FF F02D7 1519          RDRAMQ CYLDIVAM
0026D D0FF2 FFFFF 1520          RDXFR SMC,,R2
0026E 357F2 F02D8 1521          OR DB,LSBZ,RAME,R2 & NOP CYLDIVAL  ;<GET CYLINDER MULITPLICAND
0026F 337F2 1F2C0 1527          OR ZB,LSBZ,,R2 & JSR MULTPLY       ;CONVERT ABS. CYL. TO BINARY
00270 34308 FFFFF 1528          RTOR R0,R8                         ;CONVERTED CYL. LSB'S TO R8
                1529    ;
00271 323F6 FFFFF 1530  CONV13: QTOR R6                            ;CONVERTED CYL MSB'S TO R6
00272 34341 FFFFF 1531          RTOR R4,R1                         ;MOVE HEAD TO MULTIPLIER
00273 34319 33277 1532          RTOR R1,R9 & BOZ CONV14            ;IF HEAD =0, SKIP MULTIPLICATION
00274 377F2 F02D9 1534          OR DZ,LSBZ,RAME,R2 & NOP WRDSTRK   ;< 1/2 WORDS PER TRACK TO R2
00275 337F2 1F2C0 1538          OR ZB,LSBZ,,R2 & JSR MULTPLY       ;CONVERT ABS. HEAD TO BINARY
00276 34309 FFFFF 1539          RTOR R0,R9                         ;CONVERTED HEAD TO R9
                1540    ;
00277 6505C F0026 1541  CONV14: TSTRMSK R12,26                     ;CHECK "SPACE COUNT & READ DATA"
00278 433F0 33A83 1542          RESETR R0 & BNZ CONV16             ;IF NOT, CONTINUE
00279 2D0F3 F02D6 1544          SUBB DB,Q,RAME,R3 & NOP AMPREC     ;<IF RECORD NO. GREATER...
0027A 34331 352AB 1548          RTOR R3,R1 & CJMP ALB,CONV35       ;THAN MAX., COMMAND REJECT
0027B 133F1 3327F 1549          DECRB R1 & BOZ CONV15              ;IF RECORD 0, JUMP
0027C FFFFF 3327F 1550          BOZ CONV15                         ;IF RECORD 1, JUMP
                1551    ;
0027D 373F2 F02CD 1552          RDRAMB R2,RECLNTH                  ;GET RECORD LENGTH AND COMPUTE...
0027E FFFFF 1F2C0 1553          JSR MULTPLY                        ;RELATIVE WORD NUMBER
                1554    ;
0027F 377F2 F02D9 1556  CONV15: OR DZ,LSBZ,RAME,R2 & NOP WRDSTRK   ;< 1/2 WORDS PER TRACK TO R2
00280 337F2 FFFFF 1557          OR ZB,LSBZ,,R2                     ;<WORDS PER TRACK TO R2
```

. Addr Line CDC O.S. - ADDRESS CONVERSION

```
00281 29020 FFFFF  1562            SUBB AB,Q,R2,R0                 ;Q = TRANSFER LENGTH
00282 323F7 E0284  1563            QTOR R7 & JMP $+2               ;TRANSFER LENGTH TO R7
                   1564    ;
00283 37353 F0001  1566    CONV16: RDMSKB R3,1                     ;<SET RECORD NO.= 1
00284 01390 FFFFF  1570            ADD AB,B,R9,R0                  ;ADD CONVERTED HEAD + REC. NO.
00285 01380 FFFFF  1571            ADD AB,B,R8,R0                  ;ADD CONVERTED CYL.+ (HEAD + REC.)
00286 34361 31A38  1572            RTOR R6,R1 & CJMP NCARRY,CONV05 ;ADD CONVERTED CYL. MSB'S AND...
00287 0B3F1 E0238  1573            INCRB R1 & JMP CONV05           ;COMPUTE AMPERIF CYL/HD/REC.
                   1574
                   1575
00288 34301 C0009  1576    CONV20: RTOR R0,R1 & LDCT 9             ;MOVE DIVIDEND REG. TO R1
00289 45340 F03FF  1577            AND DB,B,MSK1,R0 & NOP H#3FF    ;DELETE ALL BUT 10 LSB'S
0028A 334F1 9828A  1578            OR ZB,RSQBZ,,R1 & CJPCNT EVCNTZ,$ ;MOVE DIVIDEND MSB'S
0028B 37355 F0003  1579            RDMSKB R5,3                     ;PRESET ITERATION COUNT = 3
0028C 34782 AF7FF  1580            OR AZ,LSBZ,R8,R2 & RTS          ;DOUBLE CYLINDER DIVISOR TO R2
                   1581
                   1582    ;    DIVISION ROUTINE
                   1583    ;
0028D 3434B FFFFF  1584    CONV21: RTOR R4,R11                     ;MOVE HEAD NO. TO CYL. NO.
0028E 34334 FFFFF  1585            RTOR R3,R4                      ;MOVE REC. NO. TO HEAD NO.
0028F 433F3 FFFFF  1586            RESETR R3                       ;CLEAR REC. NO.
00290 29012 FFFFF  1587    CONV22: SUBB AB,Q,R1,R2
00291 FFFFF 31A94  1588            CJMP NCARRY,$+3                 ;IF NEGATIVE NO., JUMP
00292 33FF3 FFFFF  1589            OR ZB,LSB1,,R3                  ;SHIFT 1 INTO REC. NO.
00293 323F1 E0295  1590            QTOR R1 & JMP $+2               ;MOVE DIFF. REG. TO DIVIDEND REG.
00294 337F3 FFFFF  1591            OR ZB,LSBZ,,R3                  ;SHIFT REC. NO.
00295 333F0 982A1  1592            OR ZB,B,,R0 & CJPCNT EVCNTZ,CONV25 ;IF NOT DONE, JUMP
                   1593    ;
00296 133F5 C0005  1594            DECRB R5 & LDCT 5               ;DECR. ITERATION COUNT
00297 FFFFF A37FF  1595            CRTS ALUZ                       ;IF DONE, RETURN
                   1596    ;
00298 65155 F0001  1597            TSTREQU R5,1                    ;CHECK ITERATION COUNT
00299 FFFFF 3329C  1598            BOZ $+3                         ;IF COUNT = 1, JUMP
0029A 335F1 FFFFF  1599            OR ZB,RSBZ,,R1                  ;RIGHT SHIFT DIVIDEND
0029B 34792 E028D  1600            OR AZ,LSBZ,R9,R2 & JMP CONV21   ;HEAD DIVISOR & COMPUTE HEAD
0029C 377F2 F02CD  1601            OR DZ,LSBZ,RAME,R2 & NOP RECLNTH ;GET THE RECORD LENGTH AND...
0029D 01722 4F003  1602            ADD AB,LSBZ,R2,R2 & PSHLD 3     ;POSITION IT AS...
                   1603            LSHFT2 R2,R2                    ;THE RECORD DIVISOR
0029E 01722 887FF  1603 +          ADD AB,LSBZ,R2,R2 & CLPCNT EVCNTZ
0029F FFFFF C0006  1604            LDCT 6
002A0 FFFFF E028D  1605            JMP CONV21                      ;COMPUTE RECORD NO.
                   1606    ;
002A1 FFFFF 33AA3  1607    CONV25: BNZ $+2                         ;IF DIVIDEND LSB'S NOT 0, JUMP
002A2 337F1 E0290  1608    CONV26: OR ZB,LSBZ,,R1 & JMP CONV22     ;SHIFT DIVIDEND MSB'S & CONTINUE
                   1609    ;
002A3 337F0 FFFFF  1610            OR ZB,LSBZ,,R0                  ;SHIFT DIVIDEND LSB'S
002A4 45150 F0400  1611            TSTBIT R0,400                   ;IF BIT 10 IS...
002A5 FFFFF 332A2  1612            BOZ CONV26                      ;NOT SET, SHIFT MSB'S
002A6 45340 F03FF  1613            AND DB,B,MSK1,R0 & NOP H#3FF    ;TRUNCATE R0 TO 10 BITS
002A7 33FF1 E0290  1614            OR ZB,LSB1,,R1 & JMP CONV22     ;IF BIT 10 SET, SHIFT MSB'S + 1
                   1615    ;
```

```
 Addr            Line CDC O.S. - ADDRESS CONVERSION

                 1616   ;    CONVERSION ROUTINE ERRORS
                 1617   ;
002A8 3535F F0002 1618   CONV30:  SETRBIT R15,2                    ;SET TRUNCATION FLAG
002A9 433F2 E0102 1619            RESETR R2 & JMP COMREJ1A         ;COMMAND REJECT
                 1620   ;
                 1621   CONV34:  JSR.P DOUBLE                     ;DOUBLE DRIVE NO. IF NEEDED
002AA E0403 1F125 1621 +          SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
002AB 37352 F0004 1622   CONV35:  RDMSKB R2,4                      ;SENSE: DATA VALUE NOT AS REQD.
002AC FFFFF E0102 1623            JMP COMREJ1A                     ;COMMAND REJECT
                 1624
002AD 3735E F0002 1625   CONV37:  RDMSKB R14,2                     ;SET FORCED E.F. FLAG
                 1626            JSR.P RELDRIV                    ;RELEASE & DE-SELECT THE DRIVE
002AE E0401 1F60F 1626 +          SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
002AF FFFFF 5F07D 1627            LDAR EFPROC3
002B0 34340 E006E 1628            RTOR R4,R0 & JMP EFPROCB+2       ;PROCESS FORCED E.F.
                 1629
                 1630   CONV40:  EQU $
002B1 4505E F0020 1635            TSTRBIT R14,20                   ;**SPECIAL FLAG SET?
002B2 650FD A3A05 1636            XOR DB,Q,RAME,R13 & CRTS NALUZ,LASTDRV ;THIS DRIVE = LAST DRIVE?
002B3 333F4 A3FFF 1637            OR ZB,B,,R4 & CRTS NALUZ         ;NO, MUST DO CONVERSION; LSB'S = 0?
002B4 333F3 33AB6 1638            OR ZB,B,,R3 & BNZ $+2            ;NO, JUMP; 4 MSB'S = 0?
002B5 FFFFF A37FF 1639            CRTS ALUZ                        ;YES, DO CONVERSION
                 1640   ;
002B6 5534F 8F004 1641            MASK DB,B,MSK1,R15 & POP H#4     ;RESET ABSOLUTE ADDRESS FLAG
002B7 373F8 F02FA 1652            RDRAMB R11,CSEN408               ;GET INITIAL AMPERIF CYL. MSB'S,
002B8 FFFFF 4F003 1653            PSHLD 3
                 1654            LSHFT2 R11,R11                   ;MOVE THEM INTO POSITION...
002B9 017BB 887FF 1654 +          ADD AB,LSBZ,R11,R11 & CLPCNT EVCNTZ
002BA 353FB F02FB 1655            OR DB,B,RAME,R11 & NOP CSEN409   ;MERGE THE INITIAL LSB'S
                 1656   ;
002BB 373FC F02FC 1657            RDRAMB R12,CSEN411               ;GET THE INITIAL AMPERIF HEAD
002BC FFFFF 1F1DA 1658            JSR SEEKSENS                     ;RESTORE CYLINDER & HEAD NUMBERS
002BD 373F9 F02FD 1659            RDRAMB R9,CSEN412                ;GET THE INITIAL RECORD...
002BE 3419F F0015 1660            ALDRAM R9,SEN412                 ;AND RESTORE IT
002BF 373F6 AF2F9 1661            RDREGB RAME,R6 & RTS CWORD       ;GET INITIAL WORD NO. & RETURN
                 1662
                 1663
                 1664   ;   MULTIPLICATION ROUTINE
                 1665   ;UPON ENTRY:  R2 = MULTIPLICAND
                 1666   ;             R1 = MULTIPLIER
                 1667   ;UPON EXIT:  R0 = QUOTIENT (LSB'S) Q = MSB'S
                 1668   ;REGISTERS USED:  R0,R1,R2,R5 & Q.
                 1669
002C0 433F5 FFFFF 1670   MULTPLY:  RESETR R5                       ;CLEAR ACCUMULATOR...
002C1 433F0 FFFFF 1671            RESETR R0                        ;REGISTERS
002C2 420FF FFFFF 1672            RESETQ                           ;AND Q-REG.
002C3 45151 F0001 1673   MULTPLY2: TSTBIT R1,1                      ;CHECK L.S.B. OF MULTIPLIER
```

```
 Addr            Line CDC O.S. - ADDRESS CONVERSION

002C4 335F1 332CA  1674            OR ZB,RSBZ,,R1 & BOZ MULTPLY3      ;IF L.S.B. =0, JUMP
002C5 01320 FFFFF  1675            ADD AB,B,R2,R0                     ;INCR. ACCUMULATOR LSB'S
002C6 FFFFF 31AC8  1676            CJMP NCARRY,$+2                    ;IF NO CARRY, JUMP
002C7 0B3F5 FFFFF  1677            INCRB R5                           ;ADD ONE TO MSB'S
002C8 00355 FFFFF  1678            ADD AQ,B,R5,R5                     ;INCR. ACCUMULATOR MSB'S
002C9 333F1 FFFFF  1679            OR ZB,B,,R1                        ;IS THE MULTIPLIER = 0?
002CA 336F2 33AC3  1680   MULTPLY3: OR ZB,LSBQZ,,R2 & BNZ MULTPLY2    ;SHIFT MULTIPLICAND; NO, JUMP BACK
002CB 3405F AF7FF  1681            RTOQ R5 & RTS                      ;MOVE MSB'S TO Q-REG.& RETURN
                   1682
                   1683
                   1685   TITLE CDC O.S. - CLEAR ROUTINE
                   1693
002CC 433F7 3CAFB  1694   CLR000:  RESETR R7 & CJMP NPWRONFG,CLR03    ;IF NOT POWER UP FLAG, JUMP
002CD FFFFF E02E8  1695            JMP CLR02A                         ;PROCESS POWER UP CLEAR
                   1696
                   1697   ;    MAINTENANCE PANEL CLEAR ROUTINE
                   1698
002CE 433F1 FFFFF  1699   CLR00:   RESETR R1
                   1700            JSR.P WRTCUCOM                     ;Clear Inter-C.U. byte
002CF E0403 1F22B  1700 +          SETPGE WRTCUCOM_SHR_H#B & JSRP WRTCUCOM
002D0 C3017 FFFFF  1701            RD SCP,PRGTGL,R7                   ;READ M.P. SWITCHES
002D1 433F0 33300  1702            RESETR R0 & BOZ CLR05              ;IF NO SWITCHES SET, JUMP
002D2 65057 F0001  1703            TSTRMSK R7,1                       ;CHECK FOR TOGGLE 1
002D3 FFFFF 33AE4  1704            BNZ CLR02                          ;IF NOT, JUMP
                   1705   ;
002D4 3715E F02B8  1706            MOVE MSK2,IREG & NOP H#2B8         ;CLEAR THE LAST...
002D5 FFFFF C0017  1707            LDCT 23                            ;SENSE BYTES SENT...
002D6 FFFFF 1F33D  1708            JSR CLR25                          ;TO THE CHANNEL
                   1709   ;
002D7 3714E F0093  1710            MOVE MSK1,IREG & NOP H#93          ;CLEAR ALL...
002D8 FFFFF C0014  1711            LDCT 20                            ;E.C.C. HISTORY...
002D9 FFFFF 1F33D  1712            JSR CLR25                          ;INFORMATION
                   1713   ;
002DA 3714E F0378  1714            MOVE MSK1,IREG & NOP DRFLAGS       ;CLEAR THE...
002DB FFFFF C0020  1715            LDCT 32                            ;SAVED DRIVE FLAGS...
002DC FFFFF 1F33D  1716            JSR CLR25                          ;FOR ALL DRIVES
                   1717   ;
002DD 3714E F008A  1718            MOVE MSK1,IREG & NOP BITDRV        ;CLEAR BIT FOR CURRENT DRIVE,
002DE FFFFF C0008  1719            LDCT 8                             ;SEEK ERROR BITS AND...
002DF FFFFF 1F33D  1720            JSR CLR25                          ;ERROR/USAGE OVERFLOW
                   1721   ;
002E0 3714E F00C0  1722            MOVE MSK1,IREG & NOP H#C0          ;CLEAR ALL...
002E1 FFFFF C011F  1723            LDCT 287                           ;ERROR/USAGE LOGGING...
002E2 FFFFF 1F33D  1724            JSR CLR25                          ;FOR ALL DRIVES
002E3 433F3 E02FE  1725            RESETR R3 & JMP CLR04
                   1726   ;
002E4 65057 F0007  1727   CLR02:   TSTRMSK R7,7                       ;CHECK FOR TOGGLE 7
002E5 433FF 332E8  1728            RESETR R15 & BOZ CLR02A            ;IF SET FULL MEM CLEAR
002E6 65057 F0017  1729            TSTRMSK R7,17                      ;IF TOGGLE 17, FULL MEM. CLEAR
002E7 FFFFF 33B00  1730            BNZ      CLR05                     ;IF NOT, CONTINUE
                   1731   ;
```

```
 Addr            Line CDC O.S. - CLEAR ROUTINE

                 1732    ;    FULL MEMORY CLEAR
                 1733    ;
002E8 E7C00 1F340 1734    CLR02A:   CTL OCP,OCPDSEL & JSR CLRHSSM      ;CLEAR HSSM BOARDS
002E9 3705F F000F 1735              RDMSKQ H#F
                 1736              JSR.P ISSCMD                       ;CLEAR THE OCP DISPLAY
002EA E0401 1F624 1736 +            SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
002EB FFFFF 4F064 1737              PSHLD 100
002EC 37340 88020 1738              RDREGB MSK1,R0 & CLPCNT EVCNTZ,H#20 ;DELAY FOR HSSM CLEAR; PRESET CTR.
002ED B2100 F0000 1739              WRTE HSSMA,LDMDARM & NOP 0         ;SET THE DISK BUFFER...
002EE B2000 F0000 1740              WRTE HSSMA,LDMDARL & NOP 0         ;ADDRESS TO 0000
002EF FFFFF 4F7FF 1741    CLR02B:   PSHLD H#7FF
002F0 82300 88000 1742              WRTE HSSMA,LDMDAT & CLPCNT EVCNTZ,0 ;WRITE ZEROS
002F1 133F0 FFFFF 1743              DECRB R0                           ;DECR. ITERATION COUNT
002F2 FFFFF 33AEF 1744              BNZ CLR02B                         ;IF NOT 0, CONTINUE CLEARING
002F3 E7C00 E2100 1745              CTL OCP,OCPDSEL & JMP DIAG         ;DO THE CONFIDENCE TEST
                 1746    ;
002F4 7710F F031B 1747    CLR02C:   XNOR DZ,FOUTL,ZERO,RAME & NOP H#31B ;SET DISPLAY 509...
002F5 7710F F031C 1748              XNOR DZ,FOUTL,ZERO,RAME & NOP H#31C ;TO ALL ONES...
002F6 7710F F031D 1749              XNOR DZ,FOUTL,ZERO,RAME & NOP H#31D ;FOR A LAMP TEST
                 1750              CJSR.P PWRONFG,UPITGOOD            ;IF POWER UP, JUMP
002F7 E0403 1C576 1750 +            SETPGE UPITGOOD_SHR_H#B & CJSRP PWRONFG,UPITGOOD
002F8 B1800 F0000 1754              WRTE HSSM,LDRTPTR & NOP 0          ;DEFAULT DRIVE POINTER = 0
002F9 433F4 FFFFF 1755              RESETR R4
002FA 433F3 E02FE 1756              RESETR R3 & JMP CLR04
                 1757    ;
                 1758    ;    SUB-SYSTEM CLEAR
                 1759    ;
002FB 3715E F0285 1760    CLR03:    MOVE MSK2,IREG & NOP EFCOUNT-2     ;LOC. OF E.F. COUNT
002FC FFFFF C0002 1761              LDCT 2
002FD FFFFF 1F33D 1762              JSR CLR25                          ;CLEAR THE E.F. COUNT
                 1763    ;
002FE FFFFF C0025 1764    CLR04:    LDCT H#25
002FF 3710E 1F33D 1765              MOVE ZERO,IREG & JSR CLR25         ;CLEAR ALL CURRENT SENSE BYTES
                 1766    ;
00300 3714E F0051 1771    CLR05:    MOVE MSK1,IREG & NOP SKMASK        ;CLEAR SEEK COMPLETE PENDING...
00301 433F4 C0038 1773              RESETR R4 & LDCT 56                ;AND ATTENTION PENDING FOR...
00302 433F3 1F33D 1774              RESETR R3 & JSR CLR25              ;ALL DRIVES & ALL HIO'S
                 1775    ;
00303 3714E F02B5 1776              MOVE MSK1,IREG & NOP DRONCYL       ;CLEAR THE DRIVE...
00304 FFFFF C0002 1777              LDCT 2                             ;ON CYLINDER BITS...
00305 FFFFF 1F33D 1778              JSR CLR25                          ;FOR ALL DRIVES
                 1779    ;
00306 3710F F00BA 1783              RESETRAM ILDFLG                    ;CLEAR THE ILD FLAGS
                 1784              JSR.P WAIT4SM
00307 E0403 1F235 1784 +            SETPGE WAIT4SM_SHR_H#B & JSRP WAIT4SM
```

Addr Line CDC O.S. - CLEAR ROUTINE

```
                    1785    ;
00308 E100B FFFFF   1786    CLR12:  CTL HSSM,RSTUSEL                ;DE-SELECT ANY DRIVE
00309 65053 F0041   1787            TSTRMSK R3,D#65                 ;CHECK FOR PHYSICAL DRIVE 65
0030A 9224F 33319   1788            WRTL HSSMA,LDDRBO,R4 & BOZ CLR14 ;IF IT IS, JUMP
                    1789    ;
0030B 91E3F 4F008   1790            WRTL HSSM,LDDRUSEL,R3 & PSHLD 8 ;DRIVE TO SELECT
0030C C1045 FFFFF   1791            RD HSSM,RDDRST1,R5
0030D C1045 B8310   1792            RD HSSM,RDDRST1,R5 & CJPOP EVCNTZ,$+3 ;READ DRIVE STATUS
0030E E1007 83FFF   1793            CTL HSSM,SETUSEL & CLPCNT NALUZ ;IF NONE, JUMP BACK; SELECT DRIVE
0030F C1045 9830F   1794            RD HSSM,RDDRST1,R5 & CJPCNT EVCNTZ,$ ;WAIT FOR STABLE DRIVE STATUS
00310 45055 F000C   1795            TSTRBIT R5,FAULT+SEEKERR        ;DRIVE FAULT OR SEEK ERROR?
00311 333F5 33317   1796            OR ZB,B,,R5 & BOZ CLR13         ;IF NOT, CONTINUE
                    1797    ;
00312 B2200 F0050   1798            WRTE HSSMA,LDDRBO & NOP H#50    ;BUS OUT FOR FAULT CLEAR & RTZ
00313 FFFFF 4F007   1799            PSHLD 7                         ;DELAY COUNTER
00314 E100A 887FF   1800            CTL HSSM,SETAG3 & CLPCNT EVCNTZ ;RESET FAULT & RE-ZERO THE DRIVE
00315 E1016 FFFFF   1801            CTL HSSM,RSTAG3
00316 333F5 FFFFF   1802            OR ZB,B,,R5
                    1803    ;
                    1804    CLR13:  CJSR.P NALUZ,RELDRIV            ;RELEASE & DE-SELECT DRIVE
00317 E0401 13E0F   1804 +          SETPGE RELDRIV_SHR_H#B & CJSRP NALUZ,RELDRIV
00318 0B3F3 E0308   1805            INCRB R3 & JMP CLR12            ;INCR. NEXT DRIVE NO.
                    1806    ;
00319 55047 F0010   1807    CLR14:  MASK DB,0,MSK1,R7 & NOP H#10    ;DON'T CHECK TOGGLE BIT 4
0031A 6605F F0007   1808            TSTQMSK 7                       ;CHECK TOGGLE 7 OR 17
0031B 433F9 33B28   1809            RESETR R9 & BNZ CLR14D          ;IF NOT, JUMP; FIRST DRIVE = 0
                    1810    ;
0031C 3705F F000F   1811            RDMSKQ H#F
0031D 321FF F002D   1812            QLDRAM EFALLOW                  ;ALLOW E.F'S ON ALL HAIO'S
0031E 433FC FFFFF   1816            RESETR R12
0031F 3735E F0020   1818    CLR14A: RDMSKB R14,20                   ;SET SPECIAL FLAG
00320 B1800 F0000   1819            WRTE HSSM,LDRTPTR & NOP 0       ;CLEAR DRIVE TYPE POINTER
00321 3439D FFFFF   1820            RTOR R9,R13                     ;DRIVE NO. TO R13
                    1821            JSR.P SEL01                     ;SELECT THE DRIVE
00322 E0401 1F187   1821 +          SETPGE SEL01_SHR_H#B & JSRP SEL01
00323 373F5 F000E   1822            RDRAMB R5,SEN114                ;GET DRIVE STATUS
                    1823            CJSR.P NALUZ,RELDRIV            ;IF ANY, RELEASE & DE-SELECT DRIVE
00324 E0401 13E0F   1823 +          SETPGE RELDRIV_SHR_H#B & CJSRP NALUZ,RELDRIV
00325 65059 F001F   1824            TSTRMSK R9,D#31                 ;DRIVE NO.= 31?
00326 0B3F9 33B1F   1825            INCRB R9 & BNZ CLR14A           ;IF NOT, JUMP; INCR. DRIVE NO.
00327 3710F F03E5   1826            RESETRAM LASTFLG                ;RESET LAST DRIVE FLAGS
                    1827    ;
                    1829    CLR14C: EQU $
00328 37350 F0036   1847    CLR14D: RDMSKB R0,D#54                  ;Msg. code for "CHANNEL READY"
00329 373F1 F03FF   1848            RDRAMB R1,CSW                   ;Get Diagnostic errors
0032A 45151 F000D   1849            TSTBIT R1,H#D                   ;Ignore OCP errors
```

Addr Line CDC O.S. - CLEAR ROUTINE

```
                 1850            CJSR.P ALUZ,ISSMESS             ;If no others, "CHANNEL READY"
0032B E0403 130D0 1850 +          SETPGE ISSMESS_SHR_H#B & CJSRP ALUZ,ISSMESS
                 1869    ;
                 1870    CLR15:  EQU $
0032C 3705F F000C 1872            RDMSKQ H#C                      ;ARGUMENT TO RESET OCP STUFF
                 1873             JSR.P ISSCMD                    ;DO IT
0032D E0401 1F624 1873 +          SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
                 1874
                 1875    CLR16:  EQU $
0032E 433FE FFFFF 1879            RESETR R14                      ;CLEAR R14 FLAG REGISTER
0032F 433FF FFFFF 1880            RESETR R15                      ;CLEAR R15 FLAG REGISTER
00330 E0110 FFFFF 1881            CTL SMC,RSTPONFG                ;RESET XMC POWER ON FLAG
00331 37359 F0010 1882            RDMSKB R9,10                    ;DEVICE END DELAY TIMER
00332 3419F F03E4 1883            ALDRAM R9,DLAYDE                ;SAVE D.E. DELAY TIMER FOR ENDING
00333 E008F FFFFF 1884            CTL SMC,SYSRST                  ;SYS. RESET
00334 373F0 F03FF 1885            RDRAMB R0,CSW                   ;GET CONFIDENCE TEST ERRORS
00335 E4177 FFFFF 1886            CTL HIO,RSTEI                   ;RESET ANY E.I.
00336 9340F FFFFF 1887            WRTL SCP,LDPRGD,R0              ;DISPLAY CONFIDENCE TEST ERRORS
00337 E30D0 1F340 1892            CTL     SCP,RPOWFLG & JSR CLRHSSM ;/CLEAR HSSM & OCP Power_on flag
00338 3705F F0011 1893            RDMSKQ H#11                     ;RESET TIMER DONE FLAG
                 1894             JSR.P ISSCMD
00339 E0401 1F624 1894 +          SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
0033A 433F1 FFFFF 1901            RESETR R1
                 1902             JSR.P WRTCUCOM                  ;Clear Inter-C.U. communications
0033B E0403 1F228 1902 +          SETPGE WRTCUCOM_SHR_H#B & JSRP WRTCUCOM
0033C E0090 E0002 1903            CTL SMC,UMPRST & JMP WAIT00     ;DE-SELECT HIO; START THE WAIT LOOP
                 1904
                 1905
0033D 3710D FFFFF 1906    CLR25:  MOVE ZERO,RAMI                  ;CLEAR ONE CHARACTER
0033E 0F1EE 9833D 1907            INCRD IREG,IREG & CJPCNT EVCNTZ,CLR25 ;IF NOT DONE, CLEAR NEXT CHAR.
0033F FFFFF AF7FF 1908            RTS                             ;RETURN
                 1909
                 1910
00340 E0082 FFFFF 1911    CLRHSSM: CTL SMC,HSSMRST                 ;INTERRUPT RESET
00341 E1004 FFFFF 1912             CTL HSSM,MSEQRST                ;MICRO-SEQUENCER RESET
00342 E1000 FFFFF 1913    CLRHSSMA: CTL HSSM,DIFRST                ;RESET DISK INTERFACE
00343 E1003 FFFFF 1914             CTL HSSM,MPCRST                 ;RESET CONTROL REGISTERS
00344 E1011 FFFFF 1915             CTL HSSM,RSECTLR                ;DISABLE XMCI ECC CONTROL REG.
00345 E1002 FFFFF 1916             CTL HSSM,RSDIPER                ;RESET...
00346 E1006 FFFFF 1917             CTL HSSM,RSMDBPER               ;PARITY ERROR...
00347 E100C FFFFF 1918             CTL HSSM,RSXMPER                ;LATCHES
00348 E1016 FFFFF 1919             CTL HSSM,RSTAG3                 ;RESET TAG 3 AND...
```

```
 Addr            Line CDC O.S. -- CLEAR ROUTINE

00349 E1019 FFFFF  1920            CTL HSSM,RSTAG4                  ;RESET TAG 4
0034A C2000 AF7FF  1921            RD HSSMA,RDINTR,R0 & RTS         ;RESET INTERRUPTS & RETURN
                   1922
                   1924    ;    The following routine is used in the CDC environment to reset the MDB.
                   1925    ;Registers used:  R0 & Q.
                   1926    ;
0034B E46F0 FFFFF  1927    RSTMDB:  CTL     HIO,DENCHT               ;$Disable channel tags
0034C B1F00 F002A  1928             WRTE    HSSM,LDXMCJ & NOP MDBCLR ;$Load the jump loc. and...
0034D E100D 1F52E  1929             CTL     HSSM,XMCJCMD & JSR DLAY7US ;$start the reset;  Delay
0034E FFFFF 1F535  1930             JSR     CHKBUSY                  ;$Wait for reset to complete
0034F E46F1 AF7FF  1931             CTL     HIO,ENCHT & RTS          ;$Enable chan. tags & return
                   1932
                   1935    TITLE CDC OPERATING SYSTEM
                   1943
                   1944
                   1945    ; CONVERT DRIVE NO. TO ERROR/USAGE LOG LOCATION.  UPON EXIT R2 & Q CONTAIN SMB
                   1946    ;MEMORY LOCATION OF THE 1ST WORD OF ERROR/USAGE FOR THIS DRIVE.
                   1947
00350 FFFFF 1F1F9  1948    ERRLOG:  JSR LOGICAL                      ;LOGICAL DRIVE NO. TO Q
00351 327F2 FFFFF  1949             OR ZQ,LSBZ,,R2
00352 01722 FFFFF  1950             ADD AB,LSBZ,R2,R2                ;DRIVE NO. X 8...
00353 00322 FFFFF  1951             ADD AQ,B,R2,R2                   ;+ DRIVE NO.= X 9
00354 05342 F00C0  1952             ADD DB,B,MSK1,R2 & NOP H#C0      ;COMPUTE 1ST LOG LOC. INTO R2...
00355 3402F AF7FF  1953             RTOQ R2 & RTS                    ;AND INTO Q; RETURN
```

```
                    1955
                    1956    ;          -- DISPLAY AGEING ROUTINE --
                    1957    ;
                    1958    ;    THIS ROUTINE WILL AGE 6 WORDS IN THE SMB MEMORY AND SAVE THE NEWEST WORD
                    1959    ;AT THE TOP OF THE AGEING BLOCK (7TH WORD).
                    1960    ;FUNCTION WORDS AND ADDRESS WORDS WILL NOT BE AGED UNLESS THE THUMBWHEEL
                    1961    ;SWITCHES ARE SET TO 4DX.  E.I. STATUS WORDS WILL ALWAYS BE AGED.
                    1962    ;
                    1963    ;UPON ENTRY:  R0 = DATA TO BE DISPLAYED (0-23)
                    1964    ;             R3 = DATA TO BE DISPLAYED (24-35)
                    1965    ;           IREG = LOC. OF NEWEST WORD TO BE DISPLAYED
                    1966    ;REGISTERS USED:  R0,R1,R2,R3,Q & IREG.
                    1967    ;
00356 C3032 FFFFF   1968    AGE00:    RD SCP,PGRMSEL,R2
00357 C3032 FFFFF   1969              RD SCP,PGRMSEL,R2                ;READ THUMBWHEEL SWITCHES
00358 45352 F0FF0   1970              MSKRBIT R2,H#FF0                 ;ISOLATE THE 8 MSB'S
00359 65152 F04D0   1971              TSTREQU R2,4D0                   ;SWITCHES = 4DX?
0035A 37160 33B65   1972              MOVE SMCST,ZERO & BNZ AGE02      ;IF NOT, SKIP AGEING
                    1973    ;
0035B 34302 FFFFF   1974    AGE01:    RTOR R0,R2                       ;DATA LSB'S TO R2
0035C 3705F F000F   1975              RDMSKQ H#F
0035D 061EE 4F005   1976              ADD DQ,FOUTL,IREG,IREG & PSHLD 5 ;LOC. OF OLDEST WORD TO AGE
0035E FFFFF 1F03B   1977              JSR SMBRD                        ;GET THE OLD WORD AND...
0035F 0F1EE 1F044   1978              INCRD IREG,IREG & JSR SMBWRT     ;AGE IT ONE POSITION
00360 3705F F0008   1979              RDMSKQ 8
00361 2E1EE 887FF   1980              SUBB DQ,FOUTL,IREG,IREG & CLPCNT EVCNTZ ;LOC. OF NEWER WORD
00362 3705F F0003   1981              RDMSKQ 3
00363 061EE FFFFF   1982              ADD DQ,FOUTL,IREG,IREG           ;LOC. OF NEWEST WORD
00364 34320 FFFFF   1983              RTOR R2,R0                       ;GET THE 24 LSB'S AND...
00365 3403F E0044   1984    AGE02:    RTOQ R3 & JMP SMBWRT             ;THE 12 MSB'S, WRITE THEM & RETURN
                    1985    ;
00366 3715E F0288   1986    AGESENS:  MOVE MSK2,IREG & NOP EFADDRES    ;LOC. OF ADDR. WORD DISPLAY
00367 433F3 E035B   1987              RESETR R3 & JMP AGE01            ;RESET MSB'S & AGE DISPLAY
                    1988
                    1989    ;    THIS ROUTINE WILL MULTIPLY A NUMBER BY 4.5 AND ROUND THE RESULT DOWN.
                    1990    ;
                    1991    ;UPON ENTRY:  R0 = NUMBER TO BE MULTIPLIED
                    1992    ;UPON EXIT:   R0 = RESULT
                    1993    ;             R1 = MSB'S (BITS 8-15 SHIFTED TO 12-23)
                    1994    ;REGISTERS USED:  R0 & R1
                    1995    ;
00368 34501 FFFFF   1996    FOUR5DN:  OR AZ,RSBZ,R0,R1                 ;DIVIDE THE NUMBER BY 2
00369 01700 FFFFF   1997              ADD AB,LSBZ,R0,R0                ;MULTIPLY THE NUMBER BY 4
0036A 01310 FFFFF   1998    FOUR5DNA: ADD AB,B,R1,R0                   ;ADD THE RESULTS INTO R0
0036B 34301 4F001   1999              RTOR R0,R1 & PSHLD 1             ;RESULTS TO R1 ALSO
                    2000              LSHFT2 R1,R1                     ;LEFT SHIFT 4 PLACES...
0036C 01711 887FF   2000 +            ADD AB,LSBZ,R1,R1 & CLPCNT EVCNTZ
0036D FFFFF AF7FF   2001              RTS                              ;AND RETURN
                    2002
                    2003    ;    THIS ROUTINE WILL MULTIPLY A NUMBER BY 4.5 AND ROUND THE RESULT UP.
                    2004    ;THE REGISTERS ARE THE SAME AS ABOVE.
                    2005    ;
0036E 34501 FFFFF   2006    FOUR5UP:  OR AZ,RSBZ,R0,R1                 ;DIVIDE THE NUMBER BY 2
```

```
 Addr            Line CDC OPERATING SYSTEM

0036F 45150 F0001  2007            TSTBIT R0,1                     ;IS IT AN ODD NUMBER?
00370 01700 3336A  2008            ADD AB,LSBZ,R0,R0 & BOZ FOUR5DNA ;MULTIPLY BY 4; IF NOT, JUMP
00371 09310 E0368  2009            ADD ABC,B,R1,R0 & JMP FOUR5DNA+1 ;ADD THE NUMBERS + 1 & JUMP
                   2010
                   2011
                   2013    TITLE CDC O.S. -- INTERRUPT PROCESSING
                   2021
                   2022
                   2023    ;    THIS ROUTINE PROCESSES ALLOWED INTERRUPTS FROM CERTAIN BOARDS IN THIS
                   2024    ;CONTROLLER.  ALL INTERRUPTS ARE PROCESSED ON A PRIORITY BASIS WITH ERRORS
                   2025    ;HAVING HIGHER PRIORITY THAN NORMAL COMPLETION INTERRUPTS.
                   2026    ;    IF THERE ARE NO ALLOWED INTERRUPTS THE ROUTINE WILL RETURN ON THE
                   2027    ;STACK (RTS).  WITH NORMAL COMPLETION, OR A CORRECTABLE DISK ERROR, THE ROUTINE
                   2028    ;WILL RETURN TO CALL +3.  THE ROUTINE WILL NOT RETURN FROM PROCESSING SERIOUS
                   2029    ;ERRORS (MOST PARITY ERRORS).
                   2030    ;
                   2031    ;THE ROUTINE IS CALLED WITH:  ADD DZC,B,SEQAR,R2 & JSR UIRPT (OR SMIRPT)
                   2032    ;REGISTERS USED (NO INTERRUPT):  R0,R1,R3,R4 & Q.
                   2033    ;UPON NORMAL COMPLETE RETURN:  R4 = CACHING STATUS
                   2034
                   2035
00372 3705F F0971  2036    UIRPT:    RDMSKQ 971                    ;SET STANDARD U-SIDE MASK
00373 343F0 FFFFF  2038              RTOR     R15,R0                ;$Flags to R0
00374 45150 F0002  2039              TSTBIT   R0,2                  ;$Truncation flag?
00375 FFFFF 33377  2063              BOZ UIRPT2                     ;IF NOT, CONTINUE
00376 5605F F0800  2064              RSTQBIT 800                    ;DON'T ALLOW 2-MSEC. TIME-OUT
                   2065    UIRPT2:   TSTMBIT 4,ILDFLG               ;ILD Running?
00377 373F0 F00DA  2065 +            RDRAMB   R0,ILDFLG
00378 45150 F0004  2065 +            TSTBIT   R0,4
00379 0F301 3337E  2066              ADD DZC,B,ZERO,R1 & BOZ IRPT00 ;IF NOT, JUMP; SET U-DONE MASK
0037A 5605F F0801  2067    UIRPT3:   RSTQBIT 801                    ;DON'T ALLOW E.F'S OR TIME-OUT
0037B FFFFF E037E  2068              JMP IRPT00
                   2069    ;
0037C 3705F F0130  2070    SMIRPT:   RDMSKQ 130                     ;SET STANDARD SM-SIDE MASK
0037D 433F1 FFFFF  2071              RESETR R1                      ;RESET U-DONE MASK
                   2072    ;
0037E 46310 FFFFF  2073    IRPT00:   AND DQ,B,INTR1,R0              ;CHECK FOR AN INTERRUPT
0037F 7410E A37FF  2074              NALDREG R0,IREG & CRTS ALUZ    ;IF NONE, RETURN; IRPT TO IREG
                   2075    ;
00380 3703F FFFFF  2077              RDREGQ   VPE                   ;$Prioritize the interrupt and...
00381 0E14E F0382  2082              ADD DQC,FOUTL,MSK1,IREG & NOP $+1
00382 FFFFF 6F7FF  2083              JIR                            ;JUMP TO PROCESS IT
                   2084    ;
00383 FFFFF AF7FF  2086              RTS                            ;$(0) Impossible, ignore it
```

Addr Line CDC O.S. - INTERRUPT PROCESSING

```
                2090    ;
00384 433F3 E039F 2091          RESETR R3 & JMP IRPT10          ;(1) PROCESS U-BUS PARITY
                2092    ;
00385 433F0 E03B9 2093          RESETR R0 & JMP IRPT12          ;(2) PROCESS XMC ERROR
                2094    ;
00386 C2000 E03CC 2095          RD HSSMA,RDINTR,R0 & JMP IRPT15 ;(3) PROCESS HSSM ERROR
                2096    ;
00387 FFFFF AF7FF 2097          RTS                             ;(4) IMPOSSIBLE, IGNORE IT
00388 FFFFF AF7FF 2098          RTS                             ;(5) IMPOSSIBLE, IGNORE IT
                2099    ;
00389 FFFFF E03F4 2100          JMP IRPT20                      ;(6) WMA INSTRUCTION PARITY
                2101    ;
0038A 37383 E03AE 2102          RDREGB HIOST,R3 & JMP IRPT11    ;(7) PROCESS HAIO INTERRUPT
                2103    ;
0038B FFFFF AF7FF 2104          RTS                             ;(8) IMPOSSIBLE, IGNORE IT
0038C FFFFF AF7FF 2105          RTS                             ;(9) IMPOSSIBLE, IGNORE IT
0038D FFFFF AF7FF 2106          RTS                             ;(10) IMPOSSIBLE, IGNORE IT
                2107    ;
0038E FFFFF AF7FF 2109          RTS                             ;$(11) Impossible, ignore it
                2113    ;
0038F FFFFF E03FF 2114          JMP IRPT31                      ;(12) 2-MSEC. U-RESPONSE TIME-OUT
                2115    ;
00390 FFFFF AF7FF 2116          RTS                             ;(13) IMPOSSIBLE, IGNORE IT
00391 FFFFF AF7FF 2117          RTS                             ;(14) IMPOSSIBLE, IGNORE IT
                2118    ;
00392 C2000 FFFFF 2119          RD HSSMA,RDINTR,R0              ;(15) HSSM COMPLETE
00393 45050 F0002 2120          TSTRBIT R0,2                    ;SM-DONE?
00394 41110 33398 2121          AND AB,FOUTL,R1,R0 & BOZ IRPT01B ;IF NOT, MUST BE U-DONE
00395 371FE F02E5 2122  IRPT01A: MOVE RAME,IREG & NOP IRPRTN    ;GET SM-RETURN LOC. LSB'S...
00396 371F9 F02E4 2123          MOVE RAME,IREGH & NOP IRPRTNA   ;AND THE MSB'S
00397 E1014 E039D 2124          CTL HSSM,RSTSIRPT & JMP IRPT04  ;RESET SM-DONE INTERRUPT
                2125    ;
00398 FFFFF A37FF 2126  IRPT01B: CRTS ALUZ                      ;IF U-DONE NOT EXPECTED, RETURN
00399 0B3F2 FFFFF 2127  IRPT02:  INCRB R2                       ;INCR. RETURN LOC.+ 1
0039A E1015 FFFFF 2128          CTL HSSM,RSTUIRPT               ;RESET U-DONE INTERRUPT
                2138    ;
0039B 0C12E FFFFF 2139  IRPT03:  ADD AZC,FOUTL,R2,IREG          ;INCR. RETURN LOC.+ 1
0039C 34929 FFFFF 2140          ALDREGU R2,IREGH
0039D E0082 8F7FF 2141  IRPT04:  CTL SMC,HSSMRST & POP          ;RESET INTERRUPT
0039E 373F4 6F242 2142          RDREGB RAME,R4 & JIR CASHSTAT   ;NORMAL COMPLETE RETURN
                2143    ;
                2160    ;
                2161    ;     U-bus Parity
                2162    ;
0039F 771FE F002E 2163  IRPT10:  XNOR DZ,FOUTL,RAME,IREG & NOP BITHIO ;GET THE BIT FOR THIS HIO
003A0 37350 F0080 2164          RDMSKB R0,80
```

```
 Addr            Line CDC O.S. - INTERRUPT PROCESSING

003A1 3705F F03A0  2165            RDMSKQ $-1
003A2 0613E FFFFF  2166            ADD D0,FOUTL,VPE,IREG          ;COMPUTE JUMP LOCATION
003A3 FFFFF 6F7FF  2167            JIR
                   2168    ;
003A4 335F0 FFFFF  2169            OR ZB,RSBZ,,R0                 ;SET INTERFACE CHECK...
003A5 335F0 FFFFF  2170            OR ZB,RSBZ,,R0                 ;BIT FOR...
003A6 335F0 FFFFF  2171            OR ZB,RSBZ,,R0                 ;THE CURRENT HIO
003A7 3410F F0020  2172            ALDRAM R0,SEN208               ;SAVE SENSE 8, FORMAT 2
                   2173    ;
003A8 45055 F0800  2174            TSTRBIT R5,800                 ;CHECK WRITE OP.
003A9 FFFFF 33BFC  2175            BNZ IRPT30                     ;IF SET, CHAN. PARITY ERROR
                   2176    ;
003AA 35350 F0008  2177            SETRBIT R0,8                   ;ADD CHAN. BUS IN PARITY TO...
003AB 3410F F0020  2178            ALDRAM R0,SEN208               ;SENSE 8, FORMAT 2
003AC 3535E F1000  2179            SETRBIT R14,1000               ;SET U-BUS PARITY FLAG
003AD FFFFF E03B6  2180            JMP IRPT11A+1                  ;SET COMPLETE FLAGS
                   2181    ;
                   2182    ;     HAIO Interrupt
                   2183    ;
003AE 45153 F0001  2184    IRPT11: TSTBIT R3,1                    ;CHANNEL PARITY ERROR?
003AF FFFFF 33BFC  2185            BNZ IRPT30                     ;IF SET, JUMP
003B0 45153 F0004  2186            TSTBIT R3,4                    ;E.F. INTERRUPT?
003B1 FFFFF 333F7  2187            BOZ IRPT21                     ;IF NOT, CHECK FURTHER
003B2 3535E F0002  2188            SETRBIT R14, 2                 ;SET FORCED E.F. FLAG
                   2189    ;
                   2190            JSR.P WAIT4SM                  ;WAIT FOR SM-DONE
003B3 E0403 1F235  2190 +          SETPGE WAIT4SM_SHR_H#B & JSRP WAIT4SM
                   2191            JSR.P CLREF                    ;CLEAR THE E.F.
003B4 E0403 1F583  2191 +          SETPGE CLREF_SHR_H#B & JSRP CLREF
                   2192    ;
003B5 3535F F0002  2193    IRPT11A: SETRBIT R15,2                 ;SET TRUNCATION FLAG
003B6 35355 F0020  2194            SETRBIT R5,20                  ;SET SM-COMPLETE FLAG
                   2196    IRPT11B: EQU  $                        ;$
003B7 3535E F0400  2198            SETRBIT R14,400                ;SET U-COMPLETE FLAG
003B8 FFFFF E0399  2199            JMP IRPT02                     ;RETURN
                   2200    ;
                   2201    ;     XMC ERROR PROCESSING
                   2202    ;
003B9 37363 FFFFF  2203    IRPT12: RDREGB SMCST,R3                ;READ & RESET XMC STATUS
003BA 333F3 FFFFF  2204            OR ZB,B,,R3
003BB E4408 333CA  2205            CTL HIO,HIOTGTR & BOZ IRPT13   ;IF NO STATUS, PHONY INTERRUPT
003BC 45153 F0001  2206            TSTBIT R3,1                    ;CHECK U-BUS PARITY BITS 0-17
003BD 433F4 333BF  2207            RESETR R4 & BOZ $+2            ;IF NOT, JUMP
003BE 35354 F0080  2208            SETRBIT R4,80                  ;SET BIT IN SENSE 10, FORMAT 2
                   2209    ;
003BF 45153 F0002  2210            TSTBIT R3,2                    ;CHECK U-BUS PARITY BITS 18-35
003C0 FFFFF 333C2  2211            BOZ $+2                        ;IF NOT, JUMP
003C1 35354 F0040  2212            SETRBIT R4,40                  ;SET BIT IN SENSE 10, FORMAT 2
                   2213    ;
003C2 45153 F0004  2214            TSTBIT R3,4                    ;CHECK CT-BUS PARITY
003C3 FFFFF 333C5  2215            BOZ $+2                        ;IF NOT, JUMP
```

Addr Line CDC O.S. - INTERRUPT PROCESSING

```
003C4 35354 F0020  2216            SETRBIT R4,20                    ;SET BIT IN SENSE 10, FORMAT 2
                   2217    ;
003C5 45153 F0008  2218            TSTBIT R3,8                      ;CHECK SMB PARITY ERROR
003C6 FFFFF 333F0  2219            BOZ IRPT18                       ;IF NOT, JUMP
003C7 35354 F0004  2220            SETRBIT R4,4                     ;SET BIT IN SENSE 10, FORMAT 2
003C8 3414F F0022  2221            ALDRAM R4,SEN210                 ;SAVE IT AND SEND...
003C9 FFFFF E04AD  2222            JMP READ62                       ;EQUIPMENT CHECK
                   2223    ;
003CA 5605F F0020  2224    IRPT13: RSTQBIT 20                       ;DELETE XMC ERROR AND...
003CB FFFFF E037E  2225            JMP IRPT00                       ;CHECK FOR ANY OTHER INTERRUPT
                   2226    ;
                   2227    ;   HSSM ERROR PROCESSING
                   2228    ;
003CC 4505F F0001  2229    IRPT15: TSTRBIT R15,1                    ;HEADER READ?
003CD FFFFF 13521  2230            CJSR ALUZ,WAITCON                ;IF NOT, WAIT TO CONTINUE
                   2231    ;
003CE 45050 F0007  2232            TSTRBIT R0,7                     ;CHECK FOR MOST SERIOUS ERRORS
003CF C1063 33BD4  2233            RD HSSM,RDHDSST,R3 & BNZ IRPT16  ;IF ANY, PROCESS THEM
003D0 45050 F0008  2234            TSTRBIT R0,8                     ;DISK ERROR?
003D1 333F0 33DC6  2235            OR ZB,B,,R0 & BNZ SMERR00        ;IF YES, PROCESS IT
003D2 FFFFF A37FF  2236            CRTS ALUZ                        ;IF NO ERROR, RETURN
                   2237            JPOP.P SEL35                     ;MUST BE SELF TEST ACTIVE
003D3 E0401 BF327  2237 +          SETPGE SEL35_SHR_H#B & JPOPP SEL35
                   2238    ;
003D4 433F4 33BD6  2239    IRPT16: RESETR R4 & BNZ $+2              ;IF ANY ERROR BITS, JUMP
003D5 E0082 AF7FF  2240            CTL SMC,HSSMRST & RTS            ;ELSE, RESET IRPT & RETURN
                   2241    ;
003D6 45053 F00E0  2242            TSTRBIT R3,H#E0                  ;XMC INTERFACE ERROR?
003D7 433F0 333E1  2243            RESETR R0 & BOZ IRPT16B          ;IF NOT, JUMP
                   2244    ;
003D8 45053 F0040  2245    IRPT16A: TSTRBIT R3,40                   ;INST. BUS PARITY?
003D9 E100C 333D8  2246            CTL HSSM,RSXMPER & BOZ $+2       ;IF NOT, JUMP; RESET ERROR
003DA 35354 F0002  2247            SETRBIT R4,2                     ;SENSE 10, FRMT.2=HSSM INST. PAR.
003DB 45053 F0020  2248            TSTRBIT R3,20                    ;CT-BUS PARITY?
003DC FFFFF 333DE  2249            BOZ $+2                          ;IF NOT, JUMP
003DD 35350 F0004  2250            SETRBIT R0,4                     ;SENSE 11, FRMT.2= CT-BUS PARITY
003DE 45053 F0080  2251            TSTRBIT R3,80                    ;U-BUS PARITY?
003DF FFFFF 333E1  2252            BOZ IRPT16B                      ;IF NOT, JUMP
003E0 35354 F0008  2253            SETRBIT R4,8                     ;SENSE 10, FRMT.2= U-BUS PARITY
                   2254    ;
003E1 45053 F0018  2255    IRPT16B: TSTRBIT R3,18                   ;MEMORY DATA BUFFER PARITY?
003E2 E1006 333E4  2256            CTL HSSM,RSMDBPER & BOZ $+2      ;NO, JUMP; RESET DATA BUFFER PARITY
003E3 35350 F0001  2257            SETRBIT R0,1                     ;SENSE 11, FRMT.2= HSSM BUFFER
                   2258    ;
003E4 45053 F0004  2259            TSTRBIT R3,4                     ;HSSM INTERNAL BUS PARITY?
003E5 FFFFF 333E7  2260            BOZ $+2                          ;IF NOT, JUMP
003E6 35350 F0002  2261            SETRBIT R0,2                     ;SENSE 11, FRMT.2=INTERNAL PARITY
                   2262    ;
003E7 45053 F0003  2263            TSTRBIT R3,3                     ;DISK INTERFACE PARITY?
003E8 E1002 333EA  2264            CTL HSSM,RSDIPER & BOZ $+2       ;NO, JUMP; RESET D.I.F. PARITY ERROR
003E9 35354 F0001  2265            SETRBIT R4,1                     ;SENSE 10, FRMT.2= D.I.F. PARITY
                   2266    ;
003EA 45053 F0100  2267            TSTRBIT R3,100                   ;D.I.F. FIFO ERROR?
003EB FFFFF 333EF  2268            BOZ $+4                          ;NO, JUMP
```

```
 Addr            Line CDC O.S. - INTERRUPT PROCESSING

                 2269             JSR.P DMACHK                       ;CHECK CURRENT DMA ADDRESS
003EC E0403 1F520 2269 +          SETPGE DMACHK_SHR_H#B & JSRP DMACHK
003ED 3605F F0010 2270            SETQBIT 10                         ;SET SENSE 9, FORMAT 2 =...
003EE 321FF F0021 2271            QLDRAM SEN209                      ;D.I.F. FIFO CHECK
                 2272    ;
003EF 3410F F0023 2273            ALDRAM R0,SEN211                   ;SAVE SENSE 11, FORMAT 2
003F0 3414F F0022 2274    IRPT18: ALDRAM R4,SEN210                   ;SAVE SENSE 10, FORMAT 2
003F1 E1000 E04AD 2275            CTL HSSM,DIFRST & JMP READ62       ;SEND EQUIP. CHECK
                 2276    ;
003F2 37350 F0008 2277    IRPT19: RDMSKB R0,8                        ;SENSE: HSSM BUSY TIME-OUT
003F3 433F4 E03EF 2278            RESETR R4 & JMP IRPT18-1           ;RESET SENSE 10, FORMAT 2
                 2279    ;
                 2280    ;     WMA Instruction Parity
                 2281    ;
003F4 E0088 FFFFF 2282    IRPT20: CTL SMC,WMARST                     ;WMA RESET
003F5 37354 F0002 2283            RDMSKB R4,2                        ;SET SENSE 10, FORMAT 2
003F6 E417D E03F0 2284            CTL HIO,RSTODID & JMP IRPT18       ;RESET ODR/IDR; EQUIP. CHECK
                 2285
                 2286
003F7 45153 F0100 2287    IRPT21: TSTBIT R3,100                      ;HAIO SEQUENCER ERROR?
003F8 FFFFF 3337A 2288            BOZ UIRPT3                         ;IF NOT, PHONY INTERRUPT
                 2289             SETMBIT 20,SEN209                  ;SENSE 9, FORMAT 2 = HAIO Sequencer
003F9 37350 F0020 2289 +          RDMSKB   R0,20
003FA 3410F F0021 2289 +          ALDRAM   R0,SEN209
003FB FFFFF E04AD 2290            JMP READ62                         ;SEND EQUIP. CHECK
```

```
 Addr            Line CDC O.S. - INTERRUPT PROCESSING


                 2292   ;
                 2293   ;   CHANNEL PARITY ERROR
                 2294   ;
003FC 3535E F8000 2295   IRPT30:   SETRBIT R14,8000              ;SET CHANNEL PARITY ERROR FLAG
003FD 55355 F0004 2296             RSTRBIT  R5,4                ;Reset U-inhibit_flag
003FE FFFFF E03B6 2297             JMP IRPT11A+1
                 2298   ;
                 2299   ;   U-RESPONSE TIME-OUT
                 2300   ;
                 2301   IRPT31:   EQU $
003FF 45055 F0800 2320             TSTRBIT R5, H#800            ;CHECK FOR WRITE OP
00400 FFFFF 33406 2321             BOZ IRPT33                   ;IF NOT WRITE OP., JUMP
                 2322   ;
00401 C2003 FFFFF 2327             RD HSSMA,RDINTR,R3           ;READ INTERRUPT REG.
00402 45053 F0001 2328             TSTRBIT R3,1                 ;CHECK FOR U-DONE
00403 FFFFF 33B99 2329             BNZ IRPT02                   ;IF COMPLETE, NORMAL RETURN
00404 3535F F0002 2330             SETRBIT R15,2                ;SET TRUNCATION FLAG
00405 E417D E03B7 2332             CTL HIO,RSTODID & JMP IRPT11B ;$EXIT & SET U-COMPLETE
                 2336   ;
00406 E417D E03B5 2338   IRPT33:   CTL HIO,RSTODID & JMP IRPT11A ;$RESET IDR & JUMP
                 2344
                 2345
                 2347   TITLE CDC O.S. - READ DATA
```

Addr Line CDC O.S. -- READ DATA

```
                    2356
                    2357
                    2358
                    2359    ;     READ/WRITE REGISTERS
                    2360    ;       R0-R4,Q & IREG = SCRATCH
                    2361    ;       R5 = READ/WRITE FLAG REGISTER
                    2362    ;               BITS    MEANING
                    2364    ;$               0      NOT USED
                    2365    ;$               1      NOT USED
                    2366    ;$               2      NOT USED
                    2372    ;________________3      U-SIDE WAITING (WRITE)
                    2373    ;                4      SM-INITIATED
                    2374    ;                5      SM-COMPLETE
                    2375    ;                6      END-OF-PACK
                    2376    ;________________7      U-FULL (WRITE)
                    2377    ;                8      SM-READ ACTIVE
                    2378    ;                9      RESTART U-SIDE (WRITE)
                    2379    ;               10      U-ACTIVE
                    2380    ;________________11     WRITE OP.
                    2381    ;               12      FORMAT ACTIVE
                    2382    ;               13      SM-WRITE ACTIVE
                    2384    ;$              14      NOT USED
                    2388    ;________________15     (FACTORY) DEFECT TRACK
                    2389    ;       R6 = STARTING WORD NO.
                    2390    ;       R7 = XFER. LENGTH (FROM FUNCT. WORD)
                    2391    ;       R8 = U-TRANSFER COUNT (WORDS)
                    2392    ;       R9 = CURRENT RECORD NUMBER
                    2393    ;       R10 = HAIO XFER. LENGTH (WORDS)
                    2394    ;       R11 = CURRENT CYLINDER
                    2395    ;       R12 = CURRENT HEAD
                    2396
                    2397    HSWAIT:   EQU H#02          ;JUMP TO HSSM WAIT LOOP
                    2398    MDBCLR:   EQU H#2A          ;Clear MDB sequencer
                    2399    RDHDR:    EQU H#30          ;READ HEADER JUMP
                    2400    RDDATA:   EQU H#32          ;READ DATA JUMP
                    2401    HUNLDMD:  EQU H#33          ;HSSM-TO-HAIO/MMA TRANSFER
                    2402    RDVRFY:   EQU H#37          ;READ VERIFY JUMP
                    2403    WAITOCON: EQU H#3A          ;WAIT TO CONTINUE
                    2404    WAITHD:   EQU H#36          ;ADV. HEAD WAIT-TO-CONTINUE
                    2405    MOVHDR:   EQU H#3B          ;MOVE HEADER TO FILE REGS.
                    2407
00407 433F5 1F4B0   2408    READ00:   RESETR   R5 & JSR NXTUADDR      ;$Compute U-start address
00408 3437A 1F12A   2409              RTOR     R7,R10 & JSR SEEK02    ;$Seek the drive
                    2421    ;                                         ;SM-WORD COUNT TO U-WORD COUNT
                    2422              JSR.P PATHCHK                   ;HSSM DATA PATH CONFIDENCE TEST
```

Addr Line CDC O.S. - READ DATA

```
00409 E0403 1F5A8  2422 +          SETPGE PATHCHK_SHR_H#B & JSRP PATHCHK
                   2423    ;
                   2424    ;    RESET STUFF BEFORE STARTING
                   2425    ;
                   2426    READ001:  EQU $
0040A FFFFF 1F4B7  2428           JSR      RDWRTCLR              ;$Clear some stuff
0040B 3497F F00A9  2433           ALDRAMU R7,RDLNTHM             ;SAVE INITIAL TRANSFER...
0040C 3417F F00AA  2434           ALDRAM R7,RDLNTHL              ;LENGTH FOR USAGE LOGGING
0040D 4505E F0008  2435           TSTRBIT R14,8                  ;HOME ADDRESS FLAG?
                   2436           CJSR.P NALUZ,READ70            ;IF SET, JUMP
0040E E0403 13ECD  2436 +          SETPGE READ70_SHR_H#B & CJSRP NALUZ,READ70
0040F 373F8 F02CD  2438           RDRAMB R8,RECLNTH              ;$U-XFER. COUNT = 1 RECORD
                   2457    ;
00410 4505E F0008  2458           TSTRBIT R14,8                  ;HOME ADDR. FLAG?
00411 FFFFF 33413  2459           BOZ READ02                     ;IF NOT, CONTINUE
00412 05348 F0004  2460           ADD DB,B,MSK1,R8 & NOP D#4     ;U-XFER. COUNT IS 4 WORDS LONGER
                   2462    ;
00413 343B3 1F5BA  2472    READ02:  RTOR R11,R3 & JSR WRT050       ;LOAD THE HEADER STUFF
00414 9120F FFFFF  2473           WRTL HSSM,LDSBCL,R0            ;LOAD REC. LENGTH 8 LSB'S
                   2474    ;
00415 4505F F0600  2475           TSTRBIT R15,600                ;ISOLATE SM-ERROR & NO DATA XFER.
00416 6605F F0400  2476           TSTMSK 400                     ;SM-ERROR ONLY?
00417 FFFFF 134CA  2477           CJSR ALUZ,LDSMADDR             ;IF YES, RELOAD SM-START ADDR.
                   2478    ;
00418 45055 F0010  2479    READ03:  TSTRBIT R5,10                  ;SM-INITIATED?
00419 FFFFF 33C2F  2480           BNZ READ06+1                   ;IF YES, CONTINUE READING
                   2481    ;
0041A FFFFF 1F535  2482           JSR CHKBUSY                    ;CHECK INTERFACE BUSY
0041B 91D9F FFFFF  2489           WRTL HSSM,LDDRREC,R9           ;LOAD THE RECORD NO.
0041C 4505F F0200  2491           TSTRBIT R15,200                ;NO DATA XFER. FLAG?
0041D FFFFF 33C28  2492           BNZ READ05                     ;IF SET, JUMP
                   2493    ;
0041E 4505E F0100  2494           TSTRBIT R14,100                ;READ HEADER MODE?
0041F FFFFF 33C24  2495           BNZ READ04                     ;IF YES, JUMP
00420 B1F00 F0032  2496    READ03A: WRTE HSSM,LDXMCJ & NOP RDDATA  ;LOAD READ DATA JUMP LOC.
00421 37352 F0040  2497           RDMSKB R2,40                   ;SET SENSE 16, FORMAT 2
00422 E100D FFFFF  2498           CTL HSSM,XMCJCMD               ;START THE READ
00423 E1017 E042E  2499           CTL HSSM,SETBUSY & JMP READ06  ;SET INTERFACE BUSY STATUS
                   2500    ;
00424 B1F00 F0030  2501    READ04:  WRTE HSSM,LDXMCJ & NOP RDHDR   ;LOAD READ HEADER JUMP LOC.
00425 37352 F0002  2502           RDMSKB R2,2                    ;SET SENSE 16, FORMAT 2
```

```
 Addr            Line CDC O.S. - READ DATA

00426 3535F F0001  2503            SETRBIT R15,1                    ;SET HEADER READ FLAG
00427 E100D E042E  2504            CTL HSSM,XMCJCMD & JMP READ06    ;START READ HEADER
                   2505    ;
00428 373F2 F00BA  2506    READ05: RDRAMB R2,ILDFLG                 ;GET ILD FLAGS
00429 45052 F0010  2507            TSTRBIT R2,10                    ;FORMAT-TO-VERIFY FLAG SET?
0042A FFFFF 33C20  2508            BNZ READ03A                      ;YES, JUMP
0042B B1F00 F0037  2509            WRTE HSSM,LDXMCJ & NOP RDVRFY    ;LOAD READ VERIFY JUMP LOC.
0042C 37352 F0001  2510            RDMSKB R2,1                      ;SET SENSE 16, FORMAT 2
0042D E100D FFFFF  2511            CTL HSSM,XMCJCMD                 ;START READ VERIFY
                   2512    ;
                   2513    ;   INITIATE DISK READ
                   2514    ;
0042E 3412F F0025  2515    READ06: ALDRAM R2,SEN216                 ;SAVE SENSE 16, FORMAT 2
                   2516    ;
0042F 37350 F04EC  2517            RDMSKB R0,ADVREC                 ;SM-DONE NORMAL RETURN LOC.
00430 FFFFF 1F4C0  2518            JSR SAVIRPT                      ;SAVE IT
00431 37350 F0413  2519            RDMSKB R0,READ02                 ;SM-ERROR RETURN LOC.
00432 FFFFF 1F4C6  2520            JSR SAVSMRTN                     ;SAVE IT IN CASE OF ERROR
00433 35355 F0110  2521            SETRBIT R5,110                   ;SET SM-READ ACTIVE/INITIATED
00434 37350 F4B1E  2522            RDMSKB R0,4B1E                   ;GET 50 MS.(MIN.) TIMER...
00435 FFFFF 1F4C2  2523            JSR SAVSMTIM                     ;AND SAVE IT
                   2524            JSR.P SMBOOK                     ;DO SM-BOOKKEEPING
00436 E0401 1F0D0  2524 +          SETPGE SMBOOK_SHR_H#B & JSRP SMBOOK
                   2525    ;
00437 FFFFF 1F37C  2526    READ08: JSR SMIRPT                       ;CHECK FOR SM-INTERRUPT
00438 FFFFF 1F4D8  2527            JSR SMTIMER                      ;NONE, CHECK SM-IRPT. TIMER
                   2528    ;
00439 45055 F0400  2529            TSTRBIT R5,400                   ;U-SIDE ACTIVE?
0043A FFFFF 33C52  2530            BNZ READ12                       ;IF YES, CHECK FOR DONE
0043B 373F0 F02DB  2531            RDRAMB   R0,RECSREAD             ;ANY RECORDS READ?
0043C FFFFF 33437  2532            BOZ READ08                       ;IF NOT, CHECK SM-INTERRUPT
0043D 4505E F0400  2533            TSTRBIT R14,400                  ;U-COMPLETE FLAG?
0043E FFFFF 33C37  2534            BNZ READ08                       ;IF SET, CHECK SM-INTERRUPT
                   2553    ;
                   2554    ;   INITIATE U-SIDE TRANSFER TO CHAN.
                   2555    ;
                   2556    READ10: EQU $
0043F FFFFF 1F535  2560            JSR CHKBUSY                      ;CHECK FOR INTERFACE BUSY
00440 45050 F0008  2561            TSTRBIT R0,8                     ;DIF Error?
00441 FFFFF 33C37  2562            BNZ READ08                       ;Yes, process it now
00442 FFFFF 1F4B4  2563            JSR ULENGTH                      ;LOAD U-XFER. LENGTH
00443 B1F00 F0033  2574            WRTE HSSM,LDXMCJ & NOP HUNLDMD   ;EVEN WORD START U-READ OP.
00444 373F0 F0268  2575    READ10A: RDRAMB R0,UADDRU                ;GET THE U-ADDR. MSB'S...
00445 9150F FFFFF  2576            WRTL HSSM,LDUARM,R0              ;AND LOAD THEM
00446 373F0 F0269  2577            RDRAMB R0,UADDRL                 ;GET THE U-ADDR. LSB'S...
00447 9140F FFFFF  2578            WRTL HSSM,LDUARL,R0              ;AND LOAD THEM
                   2579    ;
00448 C2001 FFFFF  2580            RD HSSMA,RDINTR,R1               ;READ INTERRUPTS
```

Addr Line CDC O.S. - READ DATA

```
00449 45051 F0010  2581            TSTRBIT R1,10                    ;DID THE INTERFACE GO BUSY?
0044A FFFFF 33C3F  2583            BNZ     READ10                   ;$Yes, must reload everything
0044B E100D FFFFF  2587            CTL HSSM,XMCJCMD                 ;START U-SIDE READ
                   2588    ;
0044C 3535F F4000  2589            SETRBIT R15,4000                 ;SET U-INITIATED FLAG
0044D 35355 F0400  2590    READ11: SETRBIT R5,400                   ;SET U-ACTIVE FLAG
0044E 37350 F0080  2592            RDMSKB  R0,80                    ;$Set 2-second (minimum)...
0044F D0FF0 FFFFF  2593            RDXFR   SMC,,R0                  ;$CDC U-response...
00450 35350 F07EB  2594            SETRBIT R0,7EB                   ;$time-out timer...
00451 FFFFF 1F4C4  2598            JSR SAVUTIM                      ;AND SAVE IT
                   2599    ;
00452 0F322 1F372  2600    READ12: ADD DZC,B,SEQAR,R2 & JSR UIRPT   ;CHECK FOR U-DONE
00453 FFFFF 1F4E6  2601            JSR UTIMER                       ;IT'S NOT; CHECK U-IRPT TIMER
00454 FFFFF E0459  2602            JMP READ14
                   2603    ;
                   2604    ;   U-READ DONE
00455 55355 F0400  2605            RSTRBIT R5,400                   ;RESET U-ACTIVE FLAG
00456 4505F F4000  2606            TSTRBIT R15,4000                 ;U-INITIATED FLAG?
                   2607            CJSR.P NALUZ,UBOOK               ;IF YES, U-SIDE BOOKKEEPING
00457 E0401 13939  2607 +          SETPGE UBOOK_SHR_H#B & CJSRP NALUZ,UBOOK
00458 FFFFF E045E  2609            JMP     READ15                   ;$
                   2629    ;
00459 4505F F4000  2630    READ14: TSTRBIT R15,4000                 ;U-INITIATED FLAG?
                   2631            CJSR.P NALUZ,UBOOK               ;IF SET, DO U-SIDE BOOKKEEPING
0045A E0401 13939  2631 +          SETPGE UBOOK_SHR_H#B & CJSRP NALUZ,UBOOK
0045B 45055 F0100  2632            TSTRBIT R5,100                   ;SM-READ ACTIVE?
0045C FFFFF 33C37  2633            BNZ READ08                       ;IF YES, CHECK FOR SM-DONE
0045D FFFFF E0452  2634            JMP READ12                       ;ELSE, CHECK FOR U-DONE
                   2635    ;
                   2636    READ15: JSR.P CHKUDMA                   ;CHECK U-SIDE DMA ADDR.
0045E E0403 1F53B  2636 +          SETPGE CHKUDMA_SHR_H#B & JSRP CHKUDMA
0045F 4505E F0400  2637            TSTRBIT R14,400                  ;U-COMPLETE?
                   2638            BNZ.P READDONE                   ;IF YES, Read is complete
00460 E0403 33E81  2638 +          SETPGE READDONE_SHR_H#B & BNZP READDONE
                   2639    ;
00461 373F1 F02CD  2640            RDRAMB  R1,RECLNTH               ;Get record length to R1
00462 34382 FFFFF  2641            RTOR    R8,R2                    ;Last U-xfer. length to R2
00463 373F0 F02DB  2642            RDRAMB  R0,RECSREAD              ;Get Records read count
00464 133F0 FFFFF  2643    READ16: DECRB   R0                       ;Decr. Records read
00465 19312 33467  2644            SUBA    AB,B,R1,R2 & BOZ $+2     ;If 0, jump; Decr. U-xfer.
00466 FFFFF 34464  2645            CJMP    AGB,READ16               ;If U-xfer.> Rec. length, READ16
00467 3410F F02DB  2646            ALDRAM  R0,RECSREAD              ;Save new Records read count
```

Addr Line CDC O.S. - READ DATA

```
                    2647    ;
00468 45055 F0100   2648            TSTRBIT R5,100                  ;SM-READ ACTIVE?
00469 FFFFF 33C37   2649            BNZ READ08                      ;IF IT IS, CHECK FOR DONE
0046A 45055 F0020   2650            TSTRBIT R5,20                   ;SM-DONE FLAG?
0046B 333F0 33C6F   2651            OR ZB,B,,R0 & BNZ READ19        ;YES, START THE U-SIDE AGAIN
0046C FFFFF 1F4C8   2652            JSR RDSMADDR                    ;ADJUST NEXT SM-ADDRESS
0046D 5535E F4000   2653            RSTRBIT R14,4000                ;RESET READ STARTED FLAG
0046E FFFFF E0413   2654            JMP READ02                      ;START NEXT DISK READ
                    2655    ;
0046F FFFFF 33C3F   2656    READ19: BNZ READ10                      ;IF ANY RECS. READ, START U-SIDE
00470 3535E F0400   2657            SETRBIT R14,400                 ;SET U-COMPLETE (EARLY END ERROR)
                    2658
00471 45055 F0100   2659    READ20: TSTRBIT R5,100                  ;SM-READ ACTIVE?
00472 FFFFF 33C37   2660            BNZ READ08                      ;IF YES, CHECK FOR DONE
00473 FFFFF E1E81   2661            JMP READDONE                    ;ELSE, READ IS COMPLETE
                    2662    ;
00474 C2000 1F531   2663    READ21: RD HSSMA,RDINTR,R0 & JSR HSSTOP$ ;STOP THE DISK READ
00475 4505E F0010   2664            TSTRBIT R14,10                  ;READ COUNT FLAG?
                    2665            BNZ.P RDCOUNT                   ;IF SET, RESET IT & JUMP
00476 E0403 33F00   2665 +          SETPGE RDCOUNT_SHR_N#B & BNZP RDCOUNT
00477 4505E F0100   2666            TSTRBIT R14,100                 ;READ HEADER MODE?
00478 FFFFF 33C91   2667            BNZ READ23                      ;IF YES, READ IS COMPLETE
00479 55355 F0010   2668            RSTRBIT R5,10                   ;RESET SM-INITIATED
0047A 45055 F0400   2669            TSTRBIT R5,400                  ;U-ACTIVE FLAG?
0047B FFFFF 33C52   2670            BNZ READ12                      ;IF YES, CHECK FOR U-DONE
0047C 4505E F0400   2671            TSTRBIT R14,400                 ;U-COMPLETE FLAG?
0047D FFFFF 33C80   2672            BNZ READ22                      ;IF YES, WE'RE DONE
0047E FFFFF 1F52E   2673            JSR DLAY7US                     ;DELAY 7 µsec. AND...
0047F FFFFF E043F   2674            JMP READ10                      ;INITIATE U-READ OP.
                    2675    ;
                    2676    ;  DATA TRANSFER COMPLETE
                    2677    ;
                    2678    READ22: EQU $
00480 FFFFF 1F525   2680            JSR     HSWAITR                 ;$HSSM Wait loop routine
00481 FFFFF 1F342   2681            JSR     CLRHSSMA                ;$Clear all HSSM boards
00482 55355 F2100   2688            RSTRBIT R5,2100                 ;RESET READ/WRITE ACTIVE
00483 3710F F0025   2689            RESETRAM SEN216                 ;CLEAR SENSE 16, FORMAT 2
00484 45055 F0800   2690            TSTRBIT R5,800                  ;WRITE-OP. FLAG?
00485 FFFFF 33C91   2691            BNZ READ23                      ;IF SET, SKIP USAGE LOGGING
                    2692    ;
00486 3715E F00A8   2693            MOVE MSK2,IREG & NOP RDLNTHM-1
00487 FFFFF 1F03B   2694            JSR SMBRD                       ;GET SAVED INITIAL XFER. LENGTH
00488 19370 1F368   2695            SUBA AB,B,R7,R0 & JSR FOUR5DN   ;SUBTRACT CUR. SM-LENGTH AND...
                    2696    ;                                       ;CONVERT TO BYTES
00489 34307 1F350   2697            RTOR R0,R7 & JSR ERRLOG         ;BYTES TO R7; LOC. OF USAGE LOG
0048A 3412E 1F03B   2698            ALDREG R2,IREG & JSR SMBRD      ;GET PREVIOUS BYTES READ AND...
0048B 01370 FFFFF   2699            ADD AB,B,R7,R0                  ;ADD THE CURRENT BYTES
0048C FFFFF 31C8E   2700            CJMP NCARRY,$+2                 ;IF NO CARRY, JUMP
0048D 0A0FF FFFFF   2701            INCRQ                           ;INCR. THE MSB'S
0048E 3412E 1F044   2702            ALDREG R2,IREG & JSR SMBWRT     ;SAVE UPDATED LOG
                    2703    ;
```

Addr Line CDC O.S. - READ DATA

```
0048F 4605F F0080  2704            TSTQBIT 80                        ;CHECK FOR 80,000,000 (HEX) BYTES
                   2705            CJSR.P NALUZ,LOGOFL               ;IF >80,000,000, LOG OVERFLOW
00490 E0404 13825  2705 +          SETPGE LOGOFL_SHR_H#B & CJSRP NALUZ,LOGOFL
                   2706    ;
                   2707    READ23: EQU $
00491 4505F F0002  2709            TSTRBIT  R15,2                    ;$Truncation bit set?
00492 FFFFF 13B4B  2710            CJSR     NALUZ,RSTMDB             ;$Yes, reset the MDB board
00493 4505E F8000  2722            TSTRBIT R14,8000                  ;CHANNEL PARITY ERROR?
00494 FFFFF 3390D  2723            BNZ PARERR1A                      ;IF YES, SEND UNIT CHECK
00495 4505E F1000  2724            TSTRBIT R14,1000                  ;U-BUS PARITY OR U-DMA ERROR?
00496 FFFFF 33CAD  2725            BNZ READ62                        ;IF YES, EQUIP. CHECK
                   2726    ;
00497 45055 F0040  2727            TSTRBIT R5,40                     ;END-OF-PACK FLAG?
00498 FFFFF 33CA9  2728            BNZ READ25                        ;IF SET, JUMP
                   2729    ;
00499 FFFFF 1F4AF  2731            JSR      DMACHKA                  ;$Check last SM-DMA address
0049A FFFFF 33CAD  2736            BNZ READ62                        ;IF CHECK FAILURE, EQUIP. CHECK
0049B 4505F F0002  2738            TSTRBIT R15,2                     ;$TRUNCATION FLAG?
0049C FFFFF 33CA7  2739            BNZ READ24                        ;$YES, EXIT NOW
                   2751    ;
                   2752    READ23B: EQU $
0049D E417D FFFFF  2754            CTL HIO,RSTODID                   ;$CLEAR ODR & IDR
0049E 4505F F8000  2767            TSTRBIT R15,8000                  ;ECC CORRECTABLE?
0049F 603FF 33092  2768            XOR AQ,B,R15,R15 & BOZ NCMPLT0    ;NO, NORMAL COMPLETE
                   2769    ;
004A0 373F0 F0007  2770            RDRAMB R0,SEN07                   ;RESTORE SENSE BYTE 7
004A1 3705F F0040  2771            RDMSKQ 40                         ;SENSE 2 = CORRECTABLE
004A2 321FF F0002  2772            QLDRAM SEN02                      ;SAVE SENSE BYTE 2
004A3 37352 F0008  2773            RDMSKB R2,8                       ;SENSE 0 = DATA CHECK
004A4 37359 F000E  2774            RDMSKB R9,H#E                     ;STATUS ACTION = E
004A5 37353 F0002  2775            RDMSKB R3,2                       ;RECOVERY ACTION = 2
004A6 433F1 E0617  2776            RESETR R1 & JMP SMERR08+1         ;RESET SENSE 1 & SEND UNIT CHECK
                   2777    ;
004A7 3735A F000A  2778    READ24: RDMSKB R10,H#A                    ;STATUS ACTION = A
004A8 E417D E0093  2779            CTL HIO,RSTODID & JMP NCMPLT1     ;NORMAL (TRUNCATION) EXIT
                   2780    ;
                   2781    READ25: SETMBIT 20,SEN01                  ;SENSE Byte 1 = End of Pack
004A9 37350 F0020  2781 +          RDMSKB   R0,20
004AA 3410F F0001  2781 +          ALDRAM   R0,SEN01
004AB 37353 F0004  2782            RDMSKB R3,4                       ;RECOVERY ACTION = 4
004AC FFFFF E0D3F  2783            JMP END10                         ;SEND UNIT CHECK
```

Addr Line CDC O.S. - READ DATA

```
                    2791    ;
004AD 37350 F0020   2792    READ62:   RDM5KB R0,20                ;SENSE 7 = C.U. ERROR
004AE FFFFF E0AFB   2793              JMP SEL24                   ;SEND EQUIP. CHECK
                    2794
                    2795
004AF FFFFF E1D20   2796    DMACHKA:  JMP DMACHK
```

Addr Line CDC O.S. - READ DATA

```
                    2798    ;
004B0 34360 1F368   2799    NXTUADDR: RTOR R6,R0 & JSR FOUR5DN          ;CONVERT WORD NO. TO BYTES
004B1 3491F F0268   2800              ALDRAMU R1,UADDRU                 ;SAVE THE MSB'S
004B2 45050 F00FF   2801              TSTRBIT R0,H#FF                   ;ISOLATE THE 8 LSB'S AND...
004B3 321FF AF269   2802              QLDREG RAME & RTS UADDRL          ;SAVE THEM FOR LATER; RETURN
                    2803    ;
004B4 34380 1F36E   2804    ULENGTH:  RTOR R8,R0 & JSR FOUR5UP          ;COMPUTE U-SIDE READ BYTE COUNT
004B5 9160F FFFFF   2814              WRTL HSSM,LDUBCL,R0               ;LOAD BYTE COUNT LSB'S
004B6 A171F AF7FF   2815              WRTU HSSM,LDUBCM,R1 & RTS         ;LOAD BYTE COUNT MSB'S & RETURN
                    2816
                    2817
004B7 E1001 1F342   2818    RDWRTCLR: CTL HSSM,DIFSELR & JSR CLRHSSMA   ;CLEAR DISK INTERFACE BOARDS
004B8 FFFFF 1F525   2819              JSR HSWAITR                       ;HSSM WAIT LOOP
004B9 B1100 F0000   2820              WRTE HSSM,LDSARM & NOP 0          ;CLEAR THE SM-...
004BA B1000 F0000   2821              WRTE HSSM,LDSARL & NOP 0          ;STARTING ADDRESS
004BB 3710F F02DB   2822              RESETRAM RECSREAD                 ;RESET RECORDS READ
004BC 3710F F0029   2823              RESETRAM RELREC                   ;RESET RELATIVE RECORD NO.
004BD 3710F F02D6   2827              RESETRAM URELREC                  ;RESET U-SIDE RELATIVE RECORD
004BE 3710F F0018   2828              RESETRAM SEN415                   ;RESET SM-ERROR RETRY COUNT
004BF 3416F AF02B   2829              ALDREG R6,RAME & RTS STRTWD       ;SAVE STARTING WORD & RETURN
                    2830    ;
004C0 3490F F02E4   2831    SAVIRPT:  ALDRAMU R0,IRPRTNA                ;SAVE SM-DONE RETURN MSB'S...
004C1 3410F AF2E5   2832              ALDREG R0,RAME & RTS IRPRTN       ;AND THE LSB'S; RETURN
                    2833    ;
004C2 3490F F02E2   2834    SAVSMTIM: ALDRAMU R0,SMTIMEA                ;SAVE SM-IRPT. TIMER MSB'S
004C3 3410F AF2DC   2835              ALDREG R0,RAME & RTS SMTIME       ;SAVE THE LSB'S & RETURN
                    2836    ;
004C4 3490F F02E3   2837    SAVUTIM:  ALDRAMU R0,UTIMEA                 ;SAVE U-IRPT. TIMER MSB'S
004C5 3410F AF2DD   2838              ALDREG R0,RAME & RTS UTIME        ;SAVE THE LSB'S & RETURN
                    2839    ;
004C6 3490F F00A8   2840    SAVSMRTN: ALDRAMU R0,SMRTNA                 ;SAVE SM-ERROR RETURN MSB'S
004C7 3410F AF0AB   2841              ALDREG R0,RAME & RTS SMRTN        ;SAVE THE LSB'S & RETURN
                    2842
                    2843    ;    THIS ROUTINE COMPUTES AND RELOADS THE SM-START ADDRESS IN CASE OF AN
                    2844    ;SM-ERROR RETRY OR CROSSING A TRACK OR CYLINDER BOUNDARY.
                    2845    ;
004C8 373F1 F0029   2846    RDSMADDR: RDRAMB R1,RELREC                  ;GET THE RELATIVE RECORD NO.
004C9 333F1 E04CC   2847              OR ZB,B,,R1 & JMP LDSMADDR+2
                    2848    ;
004CA 373F0 F0029   2849    LDSMADDR: RDRAMB R0,RELREC                  ;GET RELATIVE RECORD NO.
004CB 14301 134DA   2850              SBA AZ,B,R0,R1 & CJSR ALUZ,LDSMADRC ;IF 0, JUMP; DECR. RECORD
004CC 433F0 334D3   2851              RESETR R0 & BOZ LDSMADRB          ;IF 0, JUMP
004CD 373F2 F02CD   2852              RDRAMB R2,RECLNTH                 ;GET RECORD LENGTH
004CE 4505E F0008   2853              TSTRBIT R14,8                     ;HOME ADDRESS FLAG?
004CF 34354 334D1   2854              RTOR R5,R4 & BOZ $+2              ;IF NOT JUMP; FLAGS TO R4
004D0 05342 F0002   2855              ADD DB,B,MSK1,R2 & NOP D#2        ;REC. LENGTH IS 2 WORDS MORE
004D1 FFFFF 1F2C0   2856              JSR MULTPLY                       ;COMPUTE RELATIVE WORD NO.
004D2 34345 FFFFF   2857              RTOR R4,R5                        ;RESTORE FLAGS TO R5
                    2858    ;
004D3 FFFFF 1F368   2859    LDSMADRB: JSR FOUR5DN                       ;TIMES 4.5 = BYTE LOC.
```

Addr Line CDC O.S. -- READ DATA

```
004D4 4505E F0008  2860            TSTRBIT R14,8                ;HOME ADDR. FLAG?
004D5 E0082 334D7  2861            CTL SnC,HSSMRST & BOZ $+2    ;IF NOT, JUMP; RESET INTERRUPTS
004D6 05340 F0009  2862            ADD DB,B,MSK1,R0 & NOP D#9   ;9 MORE BYTES FOR R1 COUNT
004D7 9100F FFFFF  2863            WRTL HSSM,LDSARL,R0          ;LOAD Sn-ADDR. LSB'S AND...
004D8 A111F FFFFF  2864            WRTU HSSM,LDSARM,R1          ;THE MSB'S
004D9 31110 AF7FF  2865            OR AB,FOUTL,R1,R0 & RTS      ;CHECK IF ADDR.= 0 & RETURN
                   2866    ;
004DA 273F1 AF2D0  2867    LDSMADRC: SAB DZ,B,RAME,R1 & RTS MAXREL  ;MAX. RELATIVE REC.- 1
```

Addr Line CDC O.S. - READ DATA

```
                2869
004DB 373F0 F02DC  2870   SMTIMER:  RDRAMB R0,SMTIME                ;GET TIMER LSB'S
004DC 1410F A3ADC  2871             SBA AZ,FOUTL,R0,RAME & CRTS NALUZ,SMTIME ;IF NOT 0, RETURN; DECR. TIMER
004DD 373F0 F02E2  2872             RDRAMB R0,SMTIMEA                ;GET TIMER MSB'S
004DE 1410F A3AE2  2873             SBA AZ,FOUTL,R0,RAME & CRTS NALUZ,SMTIMEA ;IF NOT 0, RETURN; DECR MSB'S
                2874   ;
                2875   ;    SM-INTERRUPT TIME-OUT
                2876   ;
004DF 55355 F0310  2877             RSTRBIT R5,310                   ;RESET SM-ACTIVE/INITIATED FLAGS
004E0 373F2 F0018  2878             RDRAMB R2,SEN415                 ;GET RETRY COUNTER
004E1 333F2 394E3  2879             OR      ZB,B,,R2 & CJMP NOLOG,$+2 ;If normal logging, jump
                2880             BNZ.P   ILD140                   ;If any retry count, Equip. Check
004E2 E0402 33BD9  2880 +           SETPGE ILD140_SHR_H#B & BNZP ILD140
004E3 2D142 F0002  2881             SUBB DB,FOUTL,MSK1,R2 & NOP D#2  ;RETRY COUNT = 2?
                2882             CJMP.P AGB,SMTIMERA              ;IF NOT, CHECK FOR CATASTROHIC
004E4 E0404 3401D  2882 +           SETPGE SMTIMERA_SHR_H#B & CJMPP AGB,SMTIMERA
004E5 FFFFF E13D9  2883             JMP ILD140                       ;ELSE, SEND EQUIP. CHECK
                2884
                2885
004E6 373F0 F02DD  2886   UTIMER:   RDRAMB R0,UTIME                  ;GET TIMER MSB'S
004E7 1410F A3ADD  2887             SBA AZ,FOUTL,R0,RAME & CRTS NALUZ,UTIME ;IF NOT 0, RETURN; DECR. TIMER
004E8 373F0 F02E3  2888             RDRAMB R0,UTIMEA                 ;GET TIMER MSB'S
004E9 1410F A3AE3  2889             SBA AZ,FOUTL,R0,RAME & CRTS NALUZ,UTIMEA ;IF NOT 0, RETURN; DECR MSB'S
                2890   ;
                2891   ;    U-INTERRUPT TIME-OUT
                2892   ;
004EA 37350 F0025  2893   UTIMER1: RDMSKB R0,25                      ;SENSE 7 = FORMAT 2, MSG. 5
004EB FFFFF E0AF8  2894             JMP SEL24                        ;SEND EQUIP. CHECK
                2895
                2896
                2897   TITLE ADVANCE TO NEXT RECORD
```

Addr Line ADVANCE TO NEXT RECORD

```
                    2899    ;    THIS ROUTINE CONTROLS ADVANCING TO THE NEXT RECORD.  IF THE NEXT RECORD
                    2900    ;IS ON THE NEXT HEAD THE HEAD WILL BE ADVANCED.  IF THE NEXT RECORD IS ON THE
                    2901    ;NEXT CYLINDER THE DRIVE WILL BE SEEKED.  THE ACTION TAKEN IS DETERMINED BY
                    2902    ;THE JUMP LOCATION THAT WAS PREVIOUSLY SET IN BOOKKEEPING (SMBOOK25).
                    2903    ;
                    2904    ADVREC:   INCSMB RECSREAD                   ;INCR. No. of records read/written
004EC 0F3F0 F02D8   2904 +          ADD      DZC,B,RAME,R0 & NOP RECSREAD
004ED 3410F F02DB   2904 +          ALDRAM   R0,RECSREAD
004EE 55355 F2100   2905            RSTRBIT R5,2100                 ;RESET SM-ACTIVE FLAGS
004EF 5535F F0400   2906            RSTRBIT R15,400                 ;RESET SM-RETRY FLAG
004F0 373F0 F02E7   2907    ADVREC01: RDRAMB R0,SAVXMCJ             ;GET SAVED JUMP LOC. AND...
004F1 91F0F FFFFF   2908            WRTL HSSM,LDXMCJ,R0             ;RELOAD IT
004F2 371FE F02E6   2909            MOVE RAME,IREG & NOP ADVLOC     ;GET JUMP LOC. FOR NEXT RECORD
004F3 373F9 6F028   2910            RDREGB RAME,R9 & JIR NEXTREC    ;GET NEXT RECORD & JUMP
                    2911    ;
004F4 C2000 1F531   2912    ADVREC03: RD HSSMA,RDINTR,R0 & JSR HSSTOP   ;STOP DISK OPERATION
004F5 4505F F4000   2913            TSTRBIT R15,4000                ;U-INITIATED FLAG?
004F6 FFFFF 334F9   2914            BOZ $+3                         ;IF NOT, JUMP
                    2915            JSR.P UBOOK05                   ;RELOAD LAST MPC JUMP LOC...
004F7 E0401 1F161   2915 +          SETPGE UBOOK05_SHR_H#B & JSRP UBOOK05
004F8 FFFFF 1F52D   2916            JSR HSWAITR1                    ;AND DO IT
004F9 3710F F0025   2917            RESETRAM SEN216                 ;CLEAR SENSE 16, FORMAT 2
004FA 55355 F0030   2918            RSTRBIT R5,30                   ;RESET SM-INITIATED & SM-COMPLETE
004FB 373FC F0027   2919            RDRAMB R12,NEXTHD               ;GET THE NEXT HEAD
004FC 341CF F0014   2920            ALDRAM R12,SEN411               ;SAVE IT FOR SENSE
004FD 4505E F0080   2921            TSTRBIT R14,80                  ;ONE TRACK FLAG?
004FE FFFFF 134C8   2922            CJSR ALUZ,RDSMADDR              ;NO, ADJUST SM-START ADDRESS
004FF 4505F F1000   2923            TSTRBIT R15,1000                ;ADVANCE CYLINDER FLAG?
00500 603FF 33D06   2924            XOR AQ,B,R15,R15 & BNZ ADVREC04 ;IF SET, RESET IT & SEEK
00501 4505E F0200   2925            TSTRBIT R14,200                 ;ALT. TRACK FLAG?
00502 FFFFF 33D06   2926            BNZ ADVREC04                    ;YES, SEEK BACK TO DEFECTIVE
                    2927    ;
00503 4505F F0800   2928            TSTRBIT R15,800                 ;ADVANCE HEAD FLAG?
00504 603FF 13D19   2929            XOR AQ,B,R15,R15 & CJSR NALUZ,ADVHEADD ;YES, RESET IT & SET NEW HEAD
00505 FFFFF E0507   2930            JMP $+2                         ;CONTINUE
                    2931    ;
00506 FFFFF 1F514   2932    ADVREC04: JSR ADVCYL                    ;SEEK TO NEXT CYLINDER
00507 5535E F4000   2933            RSTRBIT R14,4000                ;Reset Read/Write started flag
00508 45055 F0800   2937            TSTRBIT R5,800                  ;WRITE OPERATION?
00509 E1000 33D0B   2938            CTL HSSM,DIFRST & BNZ ADVREC05  ;IF YES, JUMP
0050A FFFFF E0413   2943            JMP READ02                      ;ELSE, READ NEXT TRACK
                    2944    ;
0050B 3710F F0018   2945    ADVREC05: RESETRAM SEN415               ;RESET SM-ERROR RETRY COUNT
0050C FFFFF E0580   2947            JMP      WRT025                 ;$Write next record
                    2964    ;
```

Addr Line ADVANCE TO NEXT RECORD

```
00500 E1000 1F52D  2965   ADVREC07: CTL HSSM,DIFRST & JSR HSWAITR1     ;DIF RESET & WAIT-TO-CONTINUE
0050E FFFFF 1F52E  2966             JSR DLAY7US                       ;DELAY 7-µsec. MORE
0050F 3710F F0025  2967             RESETRAM SEN216                   ;RESET OPERATION SENSE BYTE
00510 373FC F0027  2968             RDRAMB R12,NEXTHD                 ;GET THE NEXT HEAD...
00511 341CF F0014  2969             ALDRAM R12,SEN411                 ;AND SAVE IT FOR SENSE
00512 FFFFF 1F51B  2970             JSR ADVHEADA                      ;SET THE NEXT HEAD AND...
00513 FFFFF E0475  2971             JMP READ21+1                      ;WAIT FOR NEXT U-DONE
                   2972   ;
00514 5535E F0200  2973   ADVCYL:   RSTRBIT R14,200                   ;RESET ALTERNATE TRACK FLAG
00515 5535F F0800  2974             RSTRBIT R15,800                   ;RESET ADV. HEAD FLAG
00516 373FB F0026  2975             RDRAMB R11,NEXTCYL                ;GET THE NEXT/DEF. CYLINDER
00517 FFFFF 1F1DA  2976             JSR SEEKSENS                      ;SAVE CYL./HEAD FOR THE SEEK
00518 FFFFF E012A  2977             JMP SEEK02                        ;SEEK TO NEXT OR DEFECTIVE & RTS
                   2978   ;
00519 FFFFF 4F005  2979   ADVHEADD: PSHLD 5                           ;DELAY AN ADDITIONAL 1.4 µsec...
0051A FFFFF 887FF  2980             CLPCNT EVCNTZ                     ;TO ACCOMMODATE MPC MICROCODE
0051B E1082 FFFFF  2982   ADVHEADA: CTL      HSSM,HSSMRST             ;$
                   2989   ;
0051C 5535E F4000  2990   ADVHEAD:  RSTRBIT R14,4000                  ;RESET READ/WRITE STARTED
0051D 341CF F0014  2991             ALDRAM R12,SEN411                 ;SAVE HEAD FOR SENSE
0051E 922CF 4F007  2992             WRTL HSSMA,LDDRBO,R12 & PSHLD 7   ;LOAD THE HEAD AND...
0051F E1009 887FF  2993             CTL HSSM,SETAG2 & CLPCNT EVCNTZ   ;SET IT
00520 E1013 AF7FF  2994             CTL HSSM,RSTAG2 & RTS             ;RESET TAG 2 & RETURN
                   2995
00521 FFFFF 407FF  2996   WAITCON:  PUSH
00522 FFFFF 1F535  2997             JSR CHKBUSY                       ;Check for HSSM busy
00523 B1F00 F003A  2998             WRTE HSSM,LDXMCJ & NOP WAITOCON   ;MPC WAIT TO CONTIUNE JUMP
00524 FFFFF E0529  2999             JMP HSWAITRA
                   3000   ;
00525 FFFFF 407FF  3001   HSWAITR:  PUSH
00526 FFFFF 1F535  3002             JSR CHKBUSY                       ;Check for HSSM busy
00527 B1F00 F0002  3003             WRTE HSSM,LDXMCJ & NOP HSWAIT     ;MPC WAIT LOOP JUMP
00528 FFFFF FFFFF  3004             NOP
00529 FFFFF FFFFF  3005   HSWAITRA: NOP
0052A C2000 FFFFF  3006             RD       HSSMA,RDINTR,R0          ;Check for HSSM interrupt
0052B 45150 F0010  3007             TSTBIT   R0,10                    ;Busy?
0052C FFFFF D37FF  3008             CLOOP    ALUZ                     ;No, fall thru; Yes, wait
0052D E100D FFFFF  3009   HSWAITR1: CTL HSSM,XMCJCMD                  ;Execute the HSSM command
0052E FFFFF 4F020  3010   DLAY7US:  PSHLD 32                          ;LOAD THE EVENT COUNTER
0052F FFFFF 887FF  3011             CLPCNT EVCNTZ                     ;DELAY 7 MICRO-SECS.
00530 E0082 AF7FF  3012             CTL SMC,HSSMRST & RTS             ;RESET INTERRUPTS & RETURN
                   3013   ;
00531 45050 F0010  3014   HSSTOP:   TSTRBIT R0,10                     ;HSSM BUSY?
00532 E1000 3352D  3015             CTL HSSM,DIFRST & BOZ HSWAITR1    ;IF NOT, STOP DISK OPERATION
00533 FFFFF 1F535  3016             JSR CHKBUSY$                      ;WAIT FOR NOT BUSY
00534 FFFFF 8F4F0  3017             JPOP ADVREC01$                    ;RE-EXECUTE LAST OPERATION
                   3018   ;
00535 3705F F0010  3019   CHKBUSY:  RDMSKQ 10                         ;SET BUSY MASK
00536 C2000 C07FF  3020             RD HSSMA,RDINTR,R0 & LDCT H#7FF   ;READ INTERRUPT REG.
00537 40100 B83F2  3021             AND AQ,FOUTL,R0,ZERO & CJPOP EVCNTZ,1RPT19 ;CHECK FOR BUSY
00538 C2000 93537  3022             RD HSSMA,RDINTR,R0 & CJPCNT ALUZ,$-1 ;YES, JUMP BACK; NO, FALL THRU
00539 45150 F0010  3023             TSTBIT R0,10                      ;CHECK AGAIN FOR BUSY
```

```
 Addr            Line ADVANCE TO NEXT RECORD

0053A 45340 A303F  3024            AND DB,B,MSK1,R0 & CRTS ALUZ,H#3F  ;IF NOT, DELETE GARBAGE & RETURN
0053B C2000 E0537  3025            RD HSSMA,RDINTR,R0 & JMP CHKBUSY+2 ;ELSE, CONTINUE CHECKING
                   3026
                   3027
                   3029   TITLE CDC O.S. - WRITE DATA
```

Addr Line CDC O.S. - WRITE DATA

```
                    3038
                    3039    WRTDATA:  EQU H#31          ;WRITE DATA JUMP LOC.
                    3040    HLDMD:    EQU H#34          ;HIO-TO-HSSM JUMP LOC.
                    3050
                    3051    ;     SEE READ ROUTINE FOR REGISTERS
                    3052
0053C E4201 FFFFF   3060    WRT000:   CTL HIO,ODREQS                ;SET THE HAIO...
0053D E4200 FFFFF   3061              CTL HIO,ODREQR                ;INTO 'WRITE' MODE
0053E 37355 F0800   3062              RDMSKB R5,800                 ;SET WRITE OP. FLAG
0053F 3437A 1F12A   3063              RTOR R7,R10 & JSR SEEK02      ;POSITION THE DRIVE
                    3064
                    3068              JSR.P PATHCHK                 ;HSSM DATA PATH CONFIDENCE TEST
00540 E0403 1F5A8   3068 +            SETPGE PATHCHK_SHR_H#B & JSRP PATHCHK
                    3069    ;
00541 FFFFF 1F4B7   3071    WRT000A:  JSR      RDWRTCLR           ;$
00542 373F0 F02CD   3072              RDRAMB R0,RECLNTH             ;$RECORD LENGTH...
00543 FFFFF 1F368   3073              JSR FOUR5DN                   ;$TIMES 4.5 =...
00544 3490F F00AC   3074              ALDRAMU R0,RBAU               ;$STARTING RECORD...
00545 3410F F00AD   3075              ALDRAM R0,RBAL                ;$BYTE ADDRESS
                    3076    ;
00546 343A1 FFFFF   3077              RTOR R10,R1                   ;$U-XFER LENGTH TO R1
                    3078    ;
00547 433F6 FFFFF   3079              RESETR R6                     ;$START WORD IS ALWAYS 0
00548 377F0 F02CF   3080              OR DZ,LSBZ,RAME,R0 & NOP BUFCAP ;$ 1/4 BUFFER CAPACITY...
00549 337F0 FFFFF   3132              OR ZB,LSBZ,,R0                ;TIMES 4 = BUFFER CAPACITY (R0)
0054A 29101 FFFFF   3133              SUBB AB,FOUTL,R0,R1           ;IF Q IS NOT GREATER...
0054B 3400F 35D4E   3134              RTOQ R0 & CJMP NALB,WRT005    ;THAN BUFFER CAPACITY, JUMP
0054C 34308 E054F   3136              RTOR R0,R8 & JMP WRT005+1     ;$SET UXFER.(R8) = BUFFER CAPACITY
                    3141    ;
0054D E1005 FFFFF   3142    WRT004:   CTL HSSM,WRTCONT
0054E 343AB FFFFF   3144    WRT005:   RTOR R10,R8                   ;$R8 = UXFER. COUNT
0054F 37353 F0034   3158              RDMSKB R3,HLDMD               ;NORMAL U-WRITE OPERATION
                    3181
                    3182    ;   INITIATE U-SIDE WRITE
                    3183    ;
00550 FFFFF 1F535   3184    WRT011:   JSR CHKBUSY                   ;CHECK FOR INTERFACE BUSY
```

Addr Line CDC O.S. - WRITE DATA

```
00551 34360 1F368  3185            RTOR R6,R0 & JSR FOUR5DN          ;CONVERT START WORD TO BYTES
00552 9140F FFFFF  3186            WRTL HSSM,LDUARL,R0               ;LOAD THE 8 LSB'S
00553 A151F FFFFF  3187            WRTU HSSM,LDUARM,R1               ;LOAD THE 8 MSB'S
00554 34302 FFFFF  3188            RTOR R0,R2                        ;STARTING BYTE TO R2
00555 91F3F FFFFF  3189            WRTL HSSM,LDXMCJ,R3               ;LOAD U-SIDE WRITE JUMP LOC.
                   3190    ;
00556 34380 1F368  3191            RTOR R8,R0 & JSR FOUR5DN          ;CONVERT U-XFER. COUNT TO BYTES
00557 34304 FFFFF  3192            RTOR R0,R4                        ;U-COUNT BYTES TO R4
00558 1D340 F0005  3194            SUBA DB,B,MSK1,R0 & NOP D#5       ;$DECR. BYTE COUNT BY 5...
00559 34301 4F001  3199            RTOR R0,R1 & PSHLD 1              ;AND MOVE IT TO R1
                   3200            LSHFT2 R1,R1                      ;MOVE MSB'S INTO POSITION
0055A 01711 887FF  3200 +          ADD AB,LSBZ,R1,R1 & CLPCNT EVCNTZ
                   3201    ;
0055B FFFFF 1F4B5  3202    WRT012: JSR ULENGTH+1                     ;LOAD THE U-SIDE BYTE COUNT
0055C C2001 FFFFF  3203            RD HSSMA,RDINTR,R1                ;READ INTERRUPTS
0055D 45051 F0010  3204            TSTRBIT R1,10                     ;CHECK FOR BUSY
0055E 01024 33D50  3205            ADD AB,Q,R2,R4 & BNZ WRT011       ;BYTE COUNT + START BYTE = Q
0055F 323F0 FFFFF  3206            QTOR R0                           ;SO DOES R0
00560 00701 FFFFF  3207            ADD AQ,LSBZ,R0,R1                 ;MOVE MSB'S INTO POSITION AND...
00561 01711 1F4B1  3208            ADD AB,LSBZ,R1,R1 & JSR NXTUADDR+1$ ;SAVE END U-ADDR. FOR DMA CHECK
                   3212    ;
00562 4505F F0200  3213            TSTRBIT R15,200                   ;!FORMAT FLAG?
00563 FFFFF 33567  3215            BOZ WRT013                        ;$IF NOT, START THE U-SIDE
00564 E0208 1F5BE  3225            CTL SMC,AUTOLD & JSR PADZRO       ;INITIATE PAD ZEROS MODE
00565 45055 F0400  3226            TSTRBIT R5,400                    ;U-ACTIVE FLAG?
00566 E0208 33D68  3227            CTL SMC,AUTOLD & BNZ $+2          ;YES, CONTINUE CHANNEL WRITE
                   3228    ;
00567 E100D FFFFF  3229    WRT013: CTL HSSM,XMCJCMD                  ;INITIATE CHANNEL WRITE
00568 3535F F4000  3230            SETRBIT R15,4000                  ;SET U-INITIATED FLAG
                   3231    ;
00569 35355 F0C00  3232    WRT013A: SETRBIT R5,H#C00                 ;SET U-ACTIVE & WRITE OP. FLAGS
0056A 37350 F0049  3234            RDMSKB   R0,49                    ;$Set 2-second (minimum)...
0056B D0FF0 FFFFF  3235            RDXFR    SMC,,R0                  ;$CDC U-response...
0056C 35350 F0FB6  3236            SETRBIT  R0,H#FB6                 ;$time-out timer...
0056D FFFFF 1F4C4  3240            JSR SAVUTIM                       ;AND SAVE IT
                   3241    ;
0056E 0F322 1F372  3242    WRT014: ADD DZC,B,SEQAR,R2 & JSR UIRPT    ;CHECK FOR U-SIDE DONE
0056F FFFFF 1F4E6  3243            JSR UTIMER                        ;IT'S NOT, CHECK TIMER...
00570 FFFFF E057B  3244            JMP WRT015                        ;AND JUMP
                   3245    ;
                   3246    ;   U-DONE
00571 55355 F4400  3247            RSTRBIT R5,4400                   ;RESET U-ACTIVE & ADD ONE BYTE
```

Addr Line CDC O.S. -- WRITE DATA

```
00572 37380 FFFFF  3249            RDREGB   HIOST,R0                ;$Read HAIO status
00573 45050 F0080  3254            TSTRBIT R0,80                    ;ANY DATA IN HAIO?
00574 FFFFF 33578  3255            BOZ WRT014A                      ;IF NOT, CONTINUE
00575 4505E F0400  3256            TSTRBIT R14,400                  ;U-COMPLETE?
00576 FFFFF 33578  3257            BOZ WRT014A                      ;IF NOT, CONTINUE
                   3258    ;
00577 3535F F0002  3259            SETRBIT R15,2                    ;SET TRUNCATION FLAG
00578 E4402 FFFFF  3261    WRT014A: CTL HIO,HDRVGTR                  ;$CLEAR THE U-BUS
                   3262            JSR.P CHKUDMA                    ;$CHECK THE U-SIDE DMA
00579 E0403 1F538  3262 +          SETPGE CHKUDMA_SHR_H#B & JSRP CHKUDMA
0057A FFFFF E05AF  3263            JMP WRT035
                   3274    ;
                   3275    WRT015:  EQU $
00578 4505F F4000  3280            TSTRBIT R15,4000                 ;U-INITIATED?
                   3281            CJSR.P NALUZ,UBOOK               ;YES, DO U-SIDE BOOKKEEPING
0057C E0401 13939  3281 +          SETPGE UBOOK_SHR_H#B & CJSRP NALUZ,UBOOK
0057D 45055 F2000  3282    WRT016:  TSTRBIT R5,2000                  ;SM-WRITE ACTIVE?
0057E FFFFF 33DA8  3283            BNZ WRT032                       ;IF YES, CHECK FOR DONE
0057F FFFFF E0580  3288            JMP WRT025                       ;MAYBE WRITE FIRST
                   3344
                   3345
00580 4505E F0001  3346    WRT025:  TSTRBIT R14,1                    ;WRITE NEXT RECORD FLAG?
00581 FFFFF 33D93  3347            BNZ WRT030                       ;IF SET, INITIATE WRITE
00582 45055 F0020  3352    WRT026:  TSTRBIT R5,20                    ;SM-COMPLETE?
                   3353            CJSR.P ALUZ,SMBOOK20             ;NO, CHECK FOR POSSIBLE WRITE
00583 E0403 132AC  3353 +          SETPGE SMBOOK20_SHR_H#B & CJSRP ALUZ,SMBOOK20
                   3354    ;
00584 4505E F8000  3355            TSTRBIT R14,8000                 ;CHAN. PARITY ERROR?
00585 FFFFF 33C80  3356            BNZ READ22                       ;IF YES, WRITE IS COMPLETE
00586 45055 F0400  3357    WRT027:  TSTRBIT R5,400                   ;CHECK U-ACTIVE
00587 FFFFF 33D6E  3358            BNZ WRT014                       ;IF YES, CHECK FOR DONE
00588 45055 F0200  3359            TSTRBIT R5,200                   ;RESTART U-SIDE FLAG?
00589 60355 3358C  3360            XOR AQ,B,R5,R5 & BOZ $+3         ;NO, JUMP; YES, RESET IT AND...
0058A 5535E F0001  3361            RSTRBIT R14,1                    ;RESET THE WRITE NEXT FLAG
0058B 333F6 E1B1E  3362            OR ZB,B,,R6 & JMP WRT035A+2      ;RESTART THE U-SIDE
0058C 45055 F2100  3363            TSTRBIT R5,2100                  ;SM-ACTIVE?
0058D FFFFF 33D7D  3364            BNZ WRT016                       ;IF YES, CHECK FOR IRPT.
0058E FFFFF E0593  3365            JMP WRT030                       ;ELSE, START DISK WRITE
                   3366    ;
```

Addr Line CDC O.S. - WRITE DATA

```
                3367    ;    CHANNEL DATA WAS SLOW.  WE MUST SKIP ONE DISK REV.
                3368    ;
0058F FFFFF 1F52D 3369    WRT028:   JSR HSWAITR1                    ;WAIT TO CONTINUE
00590 55355 F0010 3370              RSTRBIT R5,10                   ;RESET SM-WRITE INITIATED
00591 5535E F4000 3371              RSTRBIT R14,4000                ;RESET READ/WRITE STARTED
00592 FFFFF E056E 3372              JMP WRT014                      ;CHECK U-DONE
```

Addr Line CDC O.S. - WRITE DATA

```
                    3374
                    3375    ;   START OR CONTINUE SM-WRITE
                    3376    ;
00593 45055 F0010   3377    WRT030:   TSTRBIT R5,10                     ;WRITE INITIATED?
00594 FFFFF 33D9E   3378              BNZ WRT030A                       ;IF YES, WRITE NEXT RECORD
00595 343B3 1F5BA   3384              RTOR R11,R3 & JSR WRT050          ;LOAD THE HEADER STUFF
00596 083F0 FFFFF   3385              INCRB R0                          ;INCR. SM-BYTE COUNT
00597 4505F F0400   3386              TSTRBIT R15,400                   ;SM-ERROR FLAG SET?
00598 9120F 13CCA   3387              WRTL HSSM,LDSBCL,R0 & CJSR NALUZ,LDSMADDR ;YES, RELOAD SM-ADDR.
00599 B1F00 F0031   3388              WRTE HSSM,LDXMCJ & NOP WRTDATA    ;LOAD WRITE DATA JUMP
0059A 37350 F0008   3389              RDMSKB R0,8                       ;WRITE DATA
0059B 3410F F0025   3390              ALDRAM R0,SEN216                  ;SAVE SENSE 16, FORMAT 2
0059C E100D FFFFF   3391              CTL HSSM,XMCJCMD                  ;START THE WRITE
0059D E1017 FFFFF   3392              CTL HSSM,SETBUSY                  ;SET INTERFACE BUSY STATUS
                    3393    ;
0059E E1005 FFFFF   3394    WRT030A:  CTL HSSM,WRTCONT                  ;LOAD FLAG TO CONTINUE WRITE
0059F 5535E F0001   3395              RSTRBIT R14,1                     ;RESET WRITE NEXT FLAG
005A0 35355 F2010   3396              SETRBIT R5,2010                   ;SET SM-WRITE & INITIATED FLAGS
005A1 37350 F0593   3397              RDMSKB R0,WRT030                  ;SM-ERROR RETURN LOC.
005A2 FFFFF 1F4C6   3398              JSR SAVSMRTN                      ;SAVE IT IN CASE OF ERROR
005A3 37350 F04EC   3400              RDMSKB R0,ADVREC                  ;$SM-WRITE DONE RETURN LOC.
005A4 FFFFF 1F4C0   3404              JSR SAVIRPT                       ;SAVE IT
005A5 37350 F61A8   3405              RDMSKB R0,61A8                    ;GET SM-INTERRUPT TIMER...
005A6 FFFFF 1F4C2   3406              JSR SAVSMTIM                      ;AND SAVE IT
                    3407    ;
                    3408              JSR.P SMBOOK                      ;DO SM-BOOKKEEPING
005A7 E0401 1F0D0   3408 +            SETPGE SMBOOK_SHR_M#B & JSRP SMBOOK
                    3409    ;
005A8 0F322 1F37C   3410    WRT032:   ADD DZC,B,SEQAR,R2 & JSR SM1RPT   ;CHECK FOR SM-WRITE DONE
005A9 FFFFF 1F4DB   3411              JSR SMTIMER                       ;IT'S NOT, CHECK TIMER
                    3412    ;
005AA 45055 F0020   3413              TSTRBIT R5,20                     ;SM-COMPLETE FLAG?
005AB FFFFF 33D86   3414              BNZ WRT027                        ;YES, CHECK FOR U-ACTIVE
005AC 4505E F0001   3415              TSTRBIT R14,1                     ;WRITE NEXT FLAG?
005AD FFFFF 33582   3416              BOZ WRT026                        ;IF NOT, CHECK AGAIN
005AE FFFFF E0586   3417              JMP WRT027                        ;ELSE, CHECK FOR U-ACTIVE
                    3425    ;
                    3426    WRT035:   EQU $
005AF 4505E F0400   3431              TSTRBIT R14,400                   ;U-COMPLETE FLAG?
005B0 FFFFF 33DB6   3433              BNZ WRT037                        ;$YES, JUMP
```

```
 Addr            Line CDC O.S. - WRITE DATA

00581 4505F F4000  3437            TSTRBIT R15,4000                ;U-INITIATED?
                   3438            CJSR.P NALUZ,UBOOK              ;YES, DO U-SIDE BOOKKEEPING
005B2 E0401 13939  3438 +          SETPGE UBOOK_SHR_H#B & CJSRP NALUZ,UBOOK
005B3 4505E F0400  3439            TSTRBIT R14,400                 ;U-COMPLETE FLAG?
005B4 FFFFF 33DB6  3441            BNZ WRT037                      ;$YES, JUMP
005B5 333F6 E1B1C  3442            OR ZB,B,,R6 & JMP WRT035A
                   3462    ;
                   3463    WRT037: EQU $
005B6 E417D FFFFF  3469            CTL HIO,RSTODID                 ;RESET ODR
005B7 45055 F2100  3470            TSTRBIT R5,2100                 ;SM-ACTIVE?
005B8 FFFFF 33580  3472            BOZ WRT025                      ;$NO, CHECK FOR POSSIBLE WRITE
005B9 FFFFF E05A8  3473            JMP WRT032                      ;$ELSE, CHECK FOR SM-DONE
                   3492
                   3493
                   3494    WRT050: JSR.P DRVHDR                    ;LOAD HEADER INFO.
005BA E0403 1F569  3494 +          SETPGE DRVHDR_SHR_H#B & JSRP DRVHDR
005BB 373F0 F02CD  3495            RDRANB R0,RECLNTH               ;GET THE RECORD LENGTH AND...
005BC FFFFF 1F368  3496            JSR FOUR5DN                     ;CONVERT TO BYTES
005BD A131F AF7FF  3497            WRTU HSSM,LDSBCM,R1 & RTS       ;LOAD THE 8 MSB'S & RETURN
                   3503
                   3504
005BE 7710A FFFFF  3505    PADZRO: XNOR DZ,FOUTL,ZERO,UBR1
005BF 7710B FFFFF  3506            XNOR DZ,FOUTL,ZERO,UBR2
005C0 7710C FFFFF  3507            XNOR DZ,FOUTL,ZERO,UBR3
005C1 E4401 AF7FF  3508            CTL HIO,HDRVGTS & RTS
                   3509
                   3510    ;   THIS ROUTINE PRESETS THE STARTING RECORD BYTE REFERENCE ADDRESS TO A
                   3511    ;MINIMUM NUMBER.  THIS NUMBER IS USED TO DETERMINE WHEN ONE RECORD OF DATA
                   3512    ;HAS BEEN LOADED INTO THE MDB AND A DISK WRITE CAN BEGIN.
                   3513    ;
005C2 373F0 F02CD  3514    RBAMIN: RDRANB R0,RECLNTH               ;RECORD LENGTH...
005C3 FFFFF 1F368  3515            JSR FOUR5DN                     ;TIMES 4.5 =...
005C4 3490F F00AC  3516            ALDRANU R0,RBAU                 ;STARTING RECORD...
005C5 3410F AF0AD  3517            ALDREG R0,RAME & RTS RBAL       ;BYTE ADDRESS; RETURN
                   3518
                   3519
                   3523    TITLE CHALLENGER O.S. - SM-ERROR PROCESSING
```

Addr Line CHALLENGER O.S. - SM-ERROR PROCESSING

```
                    3526
005C6 45053 F0103   3527    SMERR00:  TSTRBIT R3,103                ;DIF PARITY OR FIFO ERROR?
005C7 433F0 335CA   3528              RESETR R0 & BOZ SMERR01       ;IF NOT, CONTINUE
005C8 E1000 FFFFF   3529              CTL HSSM,DIFRST               ;DISK INTERFACE RESET
005C9 433F4 E03D8   3530              RESETR R4 & JMP IRPT16A       ;SET UP SENSE BYTES
                    3531    ;
005CA 373F2 F0018   3532    SMERR01:  RDRAMB R2,SEN415              ;GET THE RETRY COUNT
005CB 45055 F2100   3533              TSTRBIT R5,2100               ;SM-ACTIVE?
005CC FFFFF 33DCE   3534              BNZ $+2                       ;IF YES, CONTINUE
005CD E1000 E03D5   3535              CTL HSSM,DIFRST & JMP IRPT16+1 ;ELSE, RESET PHONY IRPT & RETURN
005CE FFFFF 1F535   3536              JSR CHKBUSY                   ;Check HSSM busy
005CF C1053 FFFFF   3537              RD HSSM,RSDIFST,R3
005D0 C1053 FFFFF   3538              RD HSSM,RSDIFST,R3            ;Read Interface Status
005D1 E1000 335E2   3539              CTL HSSM,DIFRST & BOZ SMERR02 ;If none, check drive error
                    3540    ;
005D2 45055 F2000   3541              TSTRBIT R5,2000               ;Write operation?
005D3 FFFFF 13D21   3542              CJSR NALUZ,WAITCON            ;Yes, Wait-to-continue
                    3543    ;
005D4 45053 F0080   3544              TSTRBIT R3,80                 ;NO DRIVE CLOCKS?
005D5 E1000 33DFE   3545              CTL HSSM,DIFRST & BNZ SMERR05 ;IF YES, JUMP
                    3546    ;
005D6 45053 F0001   3547              TSTRBIT R3,1                  ;SYNC. ERROR?
005D7 FFFFF 33E06   3548              BNZ SMERR06                   ;IF YES, JUMP
                    3549    ;
005D8 45053 F0060   3550              TSTRBIT R3,60                 ;ECC ERROR?
                    3551              BNZ.P SMERR30                 ;IF YES, PROCESS IT
005D9 E0400 33E8A   3551 +            SETPGE SMERR30_SHR_H#B & BNZP SMERR30
                    3552    ;
005DA 45053 F0010   3553              TSTRBIT R3,10                 ;NO RECORD FOUND?
005DB FFFFF 33E32   3554              BNZ SMERR12                   ;IF YES, JUMP
                    3555    ;
005DC 45053 F0002   3556              TSTRBIT R3,2                  ;DEFECTIVE TRACK?
005DD 433F1 33E3E   3557              RESETR R1 & BNZ SMERR14       ;IF YES, JUMP
                    3558    ;
005DE 45053 F0008   3559              TSTRBIT R3,8                  ;HEAD MIS-COMPARE?
005DF FFFFF 33E1B   3560              BNZ SMERR10                   ;IF YES, JUMP
                    3561    ;
005E0 45053 F0004   3562              TSTRBIT R3,4                  ;CYLINDER MIS-COMPARE?
005E1 FFFFF 33E27   3563              BNZ SMERR11                   ;IF YES, JUMP
                    3564    ;
                    3565    ; No error status.  Check for drive error.
                    3566    ;
005E2 FFFFF 1F535   3567    SMERR02:  JSR CHKBUSY                   ;CHECK FOR BUSY
005E3 C1043 FFFFF   3568              RD HSSM,RDDRST1,R3            ;READ DRIVE STATUS...
005E4 C1043 FFFFF   3569              RD HSSM,RDDRST1,R3            ;TWICE
005E5 45053 F0010   3570              TSTRBIT R3,WRITPROT           ;WRITE PROTECT?
005E6 60333 335F4   3571              XOR AQ,B,R3,R3 & BOZ SMERR03  ;IF NOT, JUMP; RESET WRITE PROT.
005E7 45055 F0800   3572              TSTRBIT R5,800                ;WRITE OPERATION?
005E8 FFFFF 335F4   3573              BOZ SMERR03                   ;IF NOT, JUMP
005E9 373F1 F00BA   3574              RDRAMB R1,ILDFLG              ;GET ILD FLAGS
005EA 45051 F0004   3575              TSTRBIT R1,4                  ;ILD RUNNING?
005EB FFFFF 33DF0   3576              BNZ SMERR02A                  ;YES, JUMP
005EC 45053 F0008   3577              TSTRBIT R3,FAULT              ;DRIVE FAULT ALSO?
```

```
 Addr            Line CHALLENGER O.S. - SM-ERROR PROCESSING

005ED FFFFF 33DF4  3578            BNZ SMERR03                     ;YES, PROCESS IT
005EE 3413F F000E  3579            ALDRAM R3,SEN114                ;SAVE DRIVE STATUS FOR SENSE
005EF FFFFF BF112  3580            JPOP RDONLYA                    ;READ ONLY ERROR
                   3581    ;
005F0 0F322 E12A5  3582    SMERR02A: CALL R2,WRTPROT                ;PROCESS ILD WRITE PROTECT
005F1 373F2 F0018  3583            RDRAMB R2,SEN415                ;RESTORE THE RETRY COUNT
005F2 C1043 FFFFF  3584            RD HSSM,RDDRST1,R3              ;READ DRIVE STATUS
005F3 C1043 FFFFF  3585            RD HSSM,RDDRST1,R3
                   3586    ;
005F4 4505F F0001  3587    SMERR03:  TSTRBIT R15,1                  ;HEADER READ?
005F5 E1000 13D21  3588            CTL HSSM,DIFRST & CJSR NALUZ,WAITCON ;IF YES, WAIT TO CONTINUE
                   3589    ;
005F6 2D142 F0005  3590            SUBB DB,FOUTL,MSK1,R2 & NOP D#5 ;5 RETRIES?
005F7 433F0 34E6A  3591            RESETR R0 & CJMP NAGB,SMERR18   ;IF YES, PERMANENT ERROR
005F8 65053 F00A3  3592            TSTRMSK R3,SELECT+ONCYL+READY+SKEND ;NORMAL DRIVE STATUS?
                   3593            BOZ.P SMERR25+1                 ;IF YES, NOTHING WRONG, RETRY
005F9 E0400 33683  3593 +          SETPGE SMERR25+1_SHR_H#B & BOZP SMERR25+1
                   3594    ;
                   3595    ;   Drive status error
                   3596    ;
005FA E1016 395FC  3597            CTL HSSM,RSTAG3 & CJMP NOLOG,$+2 ;Normal logging, jump
005FB 34335 E0B12  3598            RTOR    R3,R5 & JMP SEL31+2     ;Otherwise, Equip. Check
005FC FFFFF 1F18B  3599            JSR     SEEK30                  ;ATTEMPT TO CORRECT ERROR
                   3600    ;
                   3601    ;   IF THE ERROR WAS CORRECTED THE PROGRAM WILL RETURN (RTS) HERE.
                   3602    ;
005FD FFFFF E0682  3603            JMP SMERR25                     ;TRY AGAIN
                   3604    ;
                   3605    ;  NO DRIVE CLOCKS.  THERE IS NOTHING WE CAN DO BUT SET UP SENSE BYTES AND
                   3606    ;SEND AN EQUIPMENT CHECK.
                   3607    ;
005FE 4505E F0020  3608    SMERR05:  TSTRBIT R14,20                 ;SPECIAL FLAG SET?
005FF FFFFF B3E00  3609            CJPOP NALUZ,$+1                 ;IF YES, POP THE STACK
00600 37351 F0004  3610            RDMSKB R1,4                     ;SENSE:  PLO FAILURE FOR...
00601 3411F F0010  3611            ALDRAM R1,SEN121                ;SENSE 21, FORMAT 1
00602 45055 F2000  3612            TSTRBIT R5,2000                 ;SM-WRITE ACTIVE?
                   3613            BOZ.P SEL23                     ;IF NOT, JUMP
00603 E0401 332FA  3613 +          SETPGE SEL23_SHR_H#B & BOZP SEL23
00604 3411F F000C  3614            ALDRAM R1,SEN112                ;SENSE 12, FORMAT 1 = PLO UNSAFE
00605 FFFFF E0AFA  3615            JMP SEL23                       ;SEND EQUIP. CHECK
                   3616    ;
                   3617    ;   SYNC. ERROR
                   3618    ;
00606 37350 F0045  3619    SMERR06:  RDMSKB R0,45                   ;SENSE 7 = COUNT NO SYNC.
00607 4505F F0020  3620            TSTRBIT R15,20                  ;NO ECC CORRECTION FLAG?
                   3621            CJSR.P NALUZ,SMERR6A            ;YES, CHECK FURTHER
00608 E0403 13F2E  3621 +          SETPGE SMERR6A_SHR_H#B & CJSRP NALUZ,SMERR6A
00609 333F2 3960C  3622            OR      ZB,B,,R2 & CJMP NOLOG,$+3 ;If Normal logging, jump
                   3623            BOZ.P   SMERR25                 ;If no retries, retry once
0060A E0400 33682  3623 +          SETPGE SMERR25_SHR_H#B & BOZP SMERR25
0060B FFFFF E060E  3624            JMP     SMERR06A                ;Otherwise, error
```

```
 Addr            Line CHALLENGER O.S. - SM-ERROR PROCESSING

0060C 2D142 F0005  3625            SUBB DB,FOUTL,MSK1,R2 & NOP D#5    ;5 RETRIES?
                   3626            CJMP.P AGB,SMERR25                 ;IF NOT, TRY AGAIN
0060D E0400 34682  3626 +          SETPGE SMERR25_SHR_H#B & CJMPP AGB,SMERR25
                   3627    ;
0060E 4505F F0001  3628    SMERR06A: TSTRBIT R15,1                     ;HEADER READ FLAG?
0060F FFFFF 33612  3629            BOZ SMERR07                        ;IF NOT, EQUIP. CHECK
00610 E1004 FFFFF  3630            CTL HSSM,MSEQRST                   ;RESET THE DISK BOARDS TO...
00611 E0082 AF7FF  3631            CTL SMC,HSSMRST & RTS              ;FORCE A TIME-OUT
                   3632    ;
00612 433F1 FFFFF  3633    SMERR07:  RESETR R1                        ;RESET SENSE BYTE 1
00613 37353 F0009  3634            RDMSKB R3,9                        ;RECOVERY ACTION = 9
00614 35351 F0080  3635            SETRBIT R1,80                      ;ADD PERM. ERROR TO SENSE 1
00615 37352 F0008  3636            RDMSKB R2,8                        ;SENSE 0 = DATA CHECK
00616 37359 F000B  3637    SMERR08:  RDMSKB R9,H#B                    ;STATUS ACTION = B
00617 3412F F0000  3638            ALDRAM R2,SEN00                    ;SAVE SENSE 0
00618 3410F F0007  3639            ALDRAM R0,SEN07                    ;SAVE SENSE 7
00619 3411F 8F001  3640            ALDREG R1,RAME & POP SEN01         ;SAVE SENSE 1
0061A E4402 E0D40  3641            CTL HIO,HDRVGTR & JMP END10A       ;SEND UNIT CHECK
                   3642    ;
                   3643    ;   HEAD MIS-COMPARE
                   3644    ;
0061B 4505F F0001  3645    SMERR10:  TSTRBIT R15,1                    ;HEADER READ FLAG?
0061C E0082 33B7C  3646            CTL SMC,HSSMRST & BNZ SM1RPT       ;IF YES, DISREGARD ERROR
                   3647    ;
0061D 2D142 F000A  3648            SUBB DB,FOUTL,MSK1,R2 & NOP D#10   ;10 RETRIES?
0061E E1016 34E73  3649            CTL HSSM,RSTAG3 & CJMP NAGB,SMERR21 ;IF YES, PERMANENT ERROR
                   3650    ;
0061F FFFFF 39E73  3651            CJMP    LOG,SMERR21                ;If Forced logging, error
00620 373FC F0014  3652            RDRAMB R12,SEN411                  ;GET CURRENT HEAD
00621 4505E F0200  3653            TSTRBIT R14,200                    ;ALT. TRACK FLAG?
00622 FFFFF 33624  3654            BOZ $+2                            ;IF NOT, JUMP
00623 373FC F00B0  3655            RDRAMB R12,ALTHEAD                 ;GET ALTERNATE HEAD
00624 FFFFF 1F51E  3656            JSR ADVHEAD+2$                     ;SET HEAD AGAIN
00625 373FC F0014  3657            RDRAMB R12,SEN411                  ;RESTORE CURRENT HEAD
00626 FFFFF E0682  3658            JMP SMERR25                        ;RETRY
                   3659    ;
                   3660    ;   CYLINDER MIS-COMPARE
                   3661    ;
00627 4505F F0001  3662    SMERR11:  TSTRBIT R15,1                    ;HEADER READ FLAG?
00628 E0082 33B7C  3663            CTL SMC,HSSMRST & BNZ SM1RPT       ;IF YES, DISREGARD THE ERROR
00629 2D142 F0002  3664            SUBB DB,FOUTL,MSK1,R2 & NOP D#2    ;2 RETRIES?
0062A 333F2 34E70  3665            OR ZB,B,,R2 & CJMP NAGB,SMERR20    ;IF YES, PERMANENT ERROR
                   3666            BOZ.P SMERR25+1                    ;NO RETRIES, IMMEDIATE RETRY
0062B E0400 33683  3666 +          SETPGE SMERR25+1_SHR_H#B & BOZP SMERR25+1
                   3667    ;
0062C FFFFF 39E70  3668            CJMP    LOG,SMERR20                ;If Forced logging, error
0062D 5535E F0200  3669            RSTRBIT R14,200                    ;RESET ALT. TRACK FLAG
0062E 0F10F F0302  3670            ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag
0062F E1016 1F186  3671            CTL HSSM,RSTAG3 & JSR SEEK22       ;RE-ZERO & RE-SEEK THE DRIVE
00630 3710F F0302  3672    SMERR11A: RESETRAM SEEKONL                 ;Clear Seek Only flag
00631 FFFFF E0682  3673            JMP SMERR25                        ;TRY READ/WRITE AGAIN
                   3674    ;
                   3675    ;   NO RECORD FOUND
```

Addr Line CHALLENGER O.S. - SM-ERROR PROCESSING

```
                3676    ;
00632 333F2 39636  3677    SMERR12:  OR ZB,B,,R2 & CJMP NOLOG,SMERR12A  ;If Normal logging, jump
                3678              BOZ.P    SMERR25                   ;If no retries, retry once
00633 E0400 33682  3678 +            SETPGE SMERR25_SHR_H#B & BOZP SMERR25
00634 37351 F0008  3679              RDMSKB   R1,8                      ;Sense 1 = No Record Found
00635 FFFFF E0639  3680              JMP      SMERR12B
00636 2D142 F0005  3681    SMERR12A: SUBB DB,FOUTL,MSK1,R2 & NOP D#5    ;5 RETRIES?
                3682              CJMP.P AGB,SMERR25                 ;IF NOT, TRY AGAIN
00637 E0400 34682  3682 +            SETPGE SMERR25_SHR_H#B & CJMPP AGB,SMERR25
00638 37351 F0088  3683              RDMSKB R1,88                       ;SENSE 1: NO REC. FOUND/PERM. ERROR
00639 37353 F000A  3684    SMERR12B: RDMSKB R3,D#10                     ;RECOVERY ACTION = 10
0063A 37350 F0045  3685              RDMSKB R0,45                       ;SENSE 7 = SYNC. ERROR
0063B 4505F F0001  3686              TSTRBIT R15,1                      ;HEADER READ?
0063C FFFFF 33E0E  3687              BNZ SMERR06A                       ;YES, JUMP
0063D 433F2 E0616  3688              RESETR R2 & JMP SMERR08            ;SEND UNIT CHECK; RESET SENSE 0
```

Addr Line CHALLENGER O.S. - SM-ERROR PROCESSING

```
                3690    ;
                3691    ;   DEFECTIVE TRACK DETECTED
                3692    ;
0063E E0082 FFFFF  3693    SMERR14:  CTL SMC,HSSMRST                  ;RESET HSSM INTERRUPTS
0063F 37160 C07FF  3694              MOVE SMCST,ZERO & LDCT H#7FF      ;LOAD A TIMER JUST IN CASE
00640 4505F F0001  3695              TSTRBIT R15,1                     ;HEADER READ?
00641 E1014 33E47  3696              CTL HSSM,RSTSIRPT & BNZ SMERR15   ;YES, GET THE ALTERNATE TRACK
00642 B1F00 F003B  3697              WRTE HSSM,LDXMCJ & NOP MOVHDR     ;MOVE THE HEADER...
00643 37350 F0647  3698              RDMSKB R0,SMERR15
00644 E100D 1F4C0  3699              CTL HSSM,XMCJCMD & JSR SAVIRPT    ;TO THE FILE REGISTERS
00645 FFFFF 1F37C  3700              JSR SMIRPT                        ;CHECK FOR MOVE COMPLETE
00646 FFFFF 98645  3701              CJPCNT EVCNTZ,$-1                 ;IT'S NOT, CHECK AGAIN
                3702    ;
00647 3710F F00A6  3703    SMERR15:  RESETRAM PARCNT                   ;RESET XMC PARITY RETRY COUNT
00648 FFFFF 5F649  3704              LDAR $+1                          ;PARITY ERROR RETRY LOC.
                3705                 JSR.P READCYL                     ;READ THE ALTERNATE CYL.
00649 E0402 1F579  3705 +            SETPGE READCYL_SHR_H#B & JSRP READCYL
                3706    ;
0064A 373F1 F02D3  3707              RDRAMB R1,MAXCYLA                 ;GET MAX. USABLE CYLINDER
0064B 373F3 F02D4  3708              RDRAMB R3,MAXCYLD                 ;GET MAX. DRIVE CYLINDER
0064C E1016 FFFFF  3709              CTL HSSM,RSTAG3                   ;CLEAR TAG 3
                3710    ;
0064D 29110 C0001  3711              SUBB AB,FOUTL,R1,R0 & LDCT 1      ;ALT. CYLINDER > MAX. USABLE?
0064E 29130 35E75  3712              SUBB AB,FOUTL,R3,R0 & CJMP NALB,SMERR22 ;NO, ERROR; IF ALT. CYL...
0064F FFFFF 35675  3713              CJMP ALB,SMERR22                  ;GREATER THAN DRIVE MAX., ERROR
                3714    ;
00650 4505E F0200  3715              TSTRBIT R14,200                   ;ALT. TRACK FLAG ALREADY SET?
00651 C1081 33656  3716              RD HSSM,RDDRFLG,R1 & BOZ SMERR17  ;NO, SET IT; READ DRIVE FLAGS
00652 C1081 FFFFF  3717              RD HSSM,RDDRFLG,R1
00653 45051 F0001  3718              TSTRBIT R1,1                      ;DEFECTIVE ALTERNATE?
00654 FFFFF 33B95  3719              BNZ IRPT01A                       ;YES, RETURN NOW
00655 FFFFF E067E  3720              JMP SMERR23                       ;ELSE, IMPROPER ALT. TRACK POINTER
                3721    ;
00656 3535E F0200  3722    SMERR17:  SETRBIT R14,200                   ;SET ALT. TRACK FLAG
00657 3410F F00AF  3723              ALDRAM R0,ALTCYL                  ;SAVE ALT. CYL. FOR SEEK
00658 C10B0 98658  3724              RD HSSM,RDDRHD,R0 & CJPCNT EVCNTZ,$ ;GET THE ALTERNATE HEAD AND...
00659 45050 F001F  3725              TSTRBIT R0,1F                     ;ISOLATE IT IN Q
0065A 321FF F00B0  3726              QLDRAM ALTHEAD                    ;SAVE IT FOR SEEK
                3727                 JSR.P XMCTPAR                     ;CHECK FOR PARITY ERROR
0065B E0403 1F35B  3727 +            SETPGE XMCTPAR_SHR_H#B & JSRP XMCTPAR
                3728                 BNZ.P HSSMPAR                     ;IF ANY, PROCESS IT
0065C E0403 33B62  3728 +            SETPGE HSSMPAR_SHR_H#B & BNZP HSSMPAR
                3729    ;
0065D 4505F F0001  3730              TSTRBIT R15,1                     ;HEADER READ?
0065E FFFFF 33662  3731              BOZ $+4                           ;NO, JUMP
0065F 37750 F8000  3732              OR DZ,LSBZ,MSK2,R0 & NOP H#8000
00660 4100E FFFFF  3733              AND AB,Q,R0,R14                   ;FIND ALTERNATE TRACK FLAG?
00661 603EE 33B95  3734              XOR AQ,B,R14,R14 & BNZ IRPT01A    ;YES, RESET IT & RETURN
00662 0F10F F0302  3735              ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag and...
00663 FFFFF 1F12A  3736              JSR SEEK02                        ;SEEK TO ALTERNATE TRACK
                3737    ;
```

```
 Addr            Line CHALLENGER O.S. - SM-ERROR PROCESSING

00664 373FB F0011  3738            RDRAMB R11,SEN408              ;GET CYLINDER MSB'S AND...
00665 017BB 4F002  3739            ADD AB,LSBZ,R11,R11 & PSHLD 2  ;MOVE THEM...
                   3740            LSHFT2 R11,R11                 ;INTO POSITION
00666 017BB 887FF  3740 +          ADD AB,LSBZ,R11,R11 & CLPCNT EVCNTZ
00667 353FB F0012  3741            OR DB,B,RAME,R11 & NOP SEN409  ;MERGE THE LSB'S
00668 3710F F0302  3742            RESETRAM SEEKONL               ;Clear Seek Only flag
00669 FFFFF E0684  3743            JMP SMERR25+2                  ;READ/WRITE THE ALT. TRACK
                   3744    ;
                   3745    ;    DISK ERROR HAS OCCURRED 5 TIMES, IT MUST BE PERMANENT.
                   3746    ;
                   3747    SMERR18: SETMBIT 80,SEN01             ;Sense 1 = Pernament Error
0066A 37350 F0080  3747 +          RDMSKB   R0,80
0066B 3410F F0001  3747 +          ALDRAM   R0,SEN01
0066C 34335 FFFFF  3748            RTOR R3,R5                     ;DRIVE STATUS TO R5
0066D 45053 F0008  3749            TSTRBIT R3,FAULT               ;DRIVE STATUS = FAULT?
                   3750            BNZ.P SEL31                    ;YES, SET DEVICE CHECK
0066E E0401 33B10  3750 +          SETPGE SEL31_SHR_H#B & BNZP SEL31
0066F FFFFF E0B12  3751            JMP SEL31+2                    ;ELSE, CHECK OTHER ERRORS
                   3752    ;
                   3753    ;    PERMANENT CYLINDER MIS-COMPARE
                   3754    ;
00670 E4402 1F1EB  3755    SMERR20: CTL HIO,HDRVGTR & JSR SKERLOG ;LOG A SEEK ERROR
00671 37350 F0016  3756            RDMSKB R0,16                   ;SET SENSE 7 = FORMAT 1, MSG. 6
00672 FFFFF E0B1D  3757            JMP SEL32A                     ;SEND EQUIP. CHECK/SEEK ERROR
                   3758    ;
                   3759    ;    PERMANENT HEAD MIS-COMPARE
                   3760    ;
00673 37350 F0017  3761    SMERR21: RDMSKB R0,17                  ;SET SENSE 7 = FORMAT 1, MSG. 7
00674 FFFFF E0B1D  3762            JMP SEL32A                     ;SEND EQUIP. CHECK/SEEK ERROR
                   3763    ;
                   3764    ;    ALTERNATE CYLINDER WAS NOT WITHIN A VALID RANGE
                   3765    ;
                   3766    SMERR22: CJMP.P   NOLOG,$+4           ;If Normal logging, jump
00675 E0400 39679  3766 +          SETPGE $+4_SHR_H#B & CJMPP NOLOG,$+4
00676 333F2 FFFFF  3767            OR       ZB,B,,R2
                   3768            BOZ.P    SMERR25               ;If no retries, retry once
00677 E0400 33682  3768 +          SETPGE SMERR25_SHR_H#B & BOZP SMERR25
00678 433F1 E067C  3769            RESETR   R1 & JMP SMERR22A     ;Otherwise, error
00679 2D142 F000A  3770            SUBB DB,FOUTL,MSK1,R2 & NOP D#10  ;IS RETRY COUNT < 10?
                   3771            CJMP.P AGB,SMERR25             ;IF YES, TRY AGAIN
0067A E0400 34682  3771 +          SETPGE SMERR25_SHR_H#B & CJMPP AGB,SMERR25
0067B 37351 F0080  3772            RDMSKB R1,80                   ;SENSE 1 = PERM. ERROR
0067C 4505F F0001  3773    SMERR22A: TSTRBIT R15,1                ;HEADER READ FLAG?
                   3774            BNZ.P IRPT01A                  ;IF YES, DISREGARD ERROR
0067D E0400 33B95  3774 +          SETPGE IRPT01A_SHR_H#B & BNZP IRPT01A
                   3775    ;
0067E 37350 F000B  3776    SMERR23: RDMSKB R0,H#B                 ;SENSE 7: IMPROPER ALT. POINTER
0067F 37352 F0002  3777            RDMSKB R2,2                    ;SENSE 0 = TRACK CONDITION CHECK
```

Addr Line CHALLENGER O.S. - SM-ERROR PROCESSING

```
00680 37353 F0003  3778            RDMSKB R3,3                    ;RECOVERY ACTION = 3
00681 FFFFF E0616  3779            JMP SMERR08                    ;SEND UNIT CHECK
```

Addr Line CHALLENGER O.S. - SM-ERROR PROCESSING

```
                    3781    ;
                    3782    ;   ATTEMPT TO RETRY ON ERROR
                    3783
00682 373F2 F0018   3784    SMERR25:  RDRAMB R2,SEN415                  ;GET THE RETRY COUNT...
00683 0C12F F0018   3785              ADD AZC,FOUTL,R2,RAME & NOP SEN415 ;AND INCR. IT
00684 55355 F2110   3786              RSTRBIT R5,2110                   ;RESET ACTIVE/INITIATE FLAGS
00685 3535F F0400   3787              SETRBIT R15,400                   ;SET SM-RETRY FLAG
00686 371FE 8F0AB   3788              MOVE RAME,IREG & POP SMRTN        ;GET THE ERROR...
00687 371F9 F00A8   3789              MOVE RAME,IREGH & NOP SMRTNA      ;RETURN LOCATION
00688 E1014 FFFFF   3790              CTL HSSM,RSTSIRPT                 ;RESET SM-DONE INTERRUPT AND...
00689 E0082 6F7FF   3791              CTL SMC,HSSMRST & JIR             ;RETRY ON ERROR
                    3792    ;
                    3793    TITLE DISK ECC ERROR PROCESSING
                    3794    ;
0068A 2D142 F0002   3795    SMERR30:  SUBB DB,FOUTL,MSK1,R2 & NOP D#2   ;RETRY COUNT < 2?
0068B FFFFF 34E91   3796              CJMP NAGB,SMERR31$                ;NO, PERMANENT ERROR
0068C 4505F F0020   3797              TSTRBIT R15,20                    ;NO ECC CORRECTION FLAG?
0068D FFFFF 33E96   3798              BNZ SMERR32$                      ;YES, JUMP
0068E 45053 F0020   3799              TSTRBIT R3,20                     ;HEADER ECC ERROR?
                    3800              BNZ.P SMERR25                     ;YES, RETRY
0068F E0400 33EB2   3800 +            SETPGE SMERR25_SHR_H#B & BNZP SMERR25
00690 FFFFF E0800   3801              JMP ECCOR00                       ;ELSE, ATTEMPT ECC CORRECTION
                    3802    ;
00691 37350 F0043   3803    SMERR31:  RDMSKB R0,43                      ;SENSE 7 = DATA ECC UNCORRECTABLE
00692 45053 F0020   3804              TSTRBIT R3,20                     ;HEADER ECC ERROR?
                    3805              BOZ.P SMERR07                     ;IF NOT, PERMANENT DATA ERROR
00693 E0400 33612   3805 +            SETPGE SMERR07_SHR_H#B & BOZP SMERR07
00694 55350 F0002   3806              RSTRBIT R0,2                      ;SENSE 7 = COUNT ECC UNCORRECTABLE
00695 FFFFF E060E   3807              JMP SMERR06A
                    3808    ;
00696 45053 F0020   3809    SMERR32:  TSTRBIT R3,20                     ;HEADER ECC ERROR?
                    3810              BOZ.P     ECCOR00                 ;NO, ATTEMPT DATA CORRECTION
00697 E0401 33000   3810 +            SETPGE ECCOR00_SHR_H#B & BOZP ECCOR00
00698 37350 F0051   3811              RDMSKB R0,51                      ;SENSE 7 = COUNT ECC CORRECTABLE
00699 FFFFF 39E9C   3812              CJMP     LOG,SMERR33              ;If Forced logging, error
0069A 333F2 FFFFF   3813    SMERR32A: OR ZB,B,,R2                       ;RETRY COUNT = 0?
                    3814              BOZ.P SMERR25                     ;YES, ONE RETRY
0069B E0400 33682   3814 +            SETPGE SMERR25_SHR_H#B & BOZP SMERR25
                    3815    ;
0069C 3535F F8000   3816    SMERR33:  SETRBIT R15,8000                  ;SET ECC CORRECTABLE FLAG
0069D 3410F F0007   3817              ALDRAM R0,SEN07                   ;SAVE SENSE 7 FOR LATER
0069E 3535E F0400   3818              SETRBIT R14,400                   ;SET U-COMPLETE FLAG
0069F 55355 F0100   3819              RSTRBIT R5,100                    ;RESET SM-ACTIVE FLAG
006A0 FFFFF E0475   3820              JMP READ21+1                      ;CHECK FOR U-DONE
                    3822    ;
00800               3823    ORG H#800
                    3825    ;
                    3826    ;   ATTEMPT ECC CORRECTION
```

Addr Line DISK ECC ERROR PROCESSING

```
                3827    ;
00800 C10E0 FFFFF  3828    ECCOR00:  RD HSSM,RDECSTA,R0
00801 C10E0 FFFFF  3829              RD HSSM,RDECSTA,R0                ;READ ECC STATUS
00802 45050 F0001  3830              TSTRBIT R0,1                      ;ERROR BIT?
                3831              BOZ.P    SMERR25                  ;IF NOT, SOMETHING'S WRONG. RETRY
00803 E0400 33682  3831 +            SETPGE SMERR25_SHR_H#B & BOZP SMERR25
00804 77301 FFFFF  3832              XNOR DZ,B,ZERO,R1                 ;PRESET SHIFT COUNT TO -1
00805 55355 F0010  3833              RSTRBIT R5,10                     ;RESET SM-INITIATED
00806 B2400 F0006  3834              WRTE HSSMA,LDECTLR & NOP H#6      ;E.C.C. "CORRECT NORMAL" MODE
00807 E1010 FFFFF  3835              CTL HSSM,SETECTLR
00808 3705F F0016  3836              RDMSKQ 16
00809 D0FF2 C0002  3837              RDXFR SMC,,R2 & LDCT 2            ;PRESET THE PERIOD (N)...
0080A 35352 F0FE9  3838              SETRBIT R2,H#FE9                  ;TO 94,185 (R2)
0080B 37354 F0001  3839              RDMSKB R4,1                       ;PRESET K = 1 (R4)
                3840    ;
0080C C10E0 FFFFF  3841              RD HSSM,RDECSTA,R0
0080D C10E0 5F00D  3842              RD HSSM,RDECSTA,R0 & LDAR $       ;READ ECC STATUS
0080E 45050 F0006  3843              TSTRBIT R0,6                      ;STATUS = E.P. OR A.E?
0080F 0B3F1 3381B  3844              INCRB R1 & BNZ ECCOR03            ;IF YES, JUMP; INCR. R1
00810 E100F 98010  3845              CTL HSSM,ECCCK & CJPCNT EVCNTZ,$  ;CLOCK ECC GENERATOR
00811 29121 C0002  3846              SUBB AB,FOUTL,R2,R1 & LDCT 2      ;IS R1 LESS THAN 94185?
00812 C10E0 74813  3847              RD HSSM,RDECSTA,R0 & CJARB NAGB,ECCOR02 ;IF YES, JUMP BACK
                3848    ;
                3849    ;   DISK ECC UNCORRECTABLE
                3850    ;
00813 37352 F009C  3851    ECCOR02:  RDMSKB R2,9C                      ;UNCORRECTABLE HISTORY LOC.
00814 FFFFF 1F099  3852              JSR ECCLOG                        ;LOG THIS ERROR FOR DISPLAY
00815 373F0 F0018  3853              RDRAMB R0,SEN415                  ;IF SM-RETRY COUNT IS 0...
00816 E1011 130BF  3854              CTL HSSM,RSECTLR & CJSR ALUZ,UNCORLOG ;INCR. UNCORRECTABLE COUNT
00817 37350 F0043  3855              RDMSKB R0,43                      ;SENSE 7 = DATA ECC UNCORRECTABLE
00818 4505F F0020  3856              TSTRBIT R15,20                    ;NO ECC CORRECTION FLAG?
                3857              BNZ.P    SMERR33                  ;YES, SET CORRECTABLE FLAG
00819 E0400 33E9C  3857 +            SETPGE SMERR33_SHR_H#B & BNZP SMERR33
0081A FFFFF E0682  3858              JMP SMERR25                       ;ELSE, RETRY THE READ
                3859    ;
0081B 45050 F0002  3860    ECCOR03:  TSTRBIT R0,2                      ;A.E. (ALIGNMENT EXCEPTION) SET?
0081C 34323 33813  3861              RTOR R2,R3 & BNZ ECCOR02          ;IF IT IS, BAD STATUS, JUMP
                3862    ;
0081D 01711 FFFFF  3863              ADD AB,LSBZ,R1,R1                 ;MULTIPLY R1 X 4...
0081E 337F1 FFFFF  3864              OR ZB,LSBZ,,R1                    ; X 2 = X8
0081F 05341 F0005  3865              ADD DB,B,MSK1,R1 & NOP D#5        ;INCR.+ 5
00820 29021 FFFFF  3866    ECCOR04:  SUBB AB,Q,R2,R1                   ;N - R1 = L (Q)
00821 FFFFF 31025  3867              CJMP CARRY,ECCOR05                ;IF L IS POSITIVE, JUMP
00822 2D144 F0008  3868              SUBB DB,FOUTL,MSK1,R4 & NOP D#8   ;IS K > 8?
00823 01332 35013  3869              ADD AB,B,R3,R2 & CJMP ALB,ECCOR02 ;IF YES, UNCORR.; INCR. N + 94185
00824 0B3F4 E0820  3870              INCRB R4 & JMP ECCOR04            ;INCR. K + 1 & JUMP BACK
                3871    ;
00825 3495F F00A5  3872    ECCOR05:  ALDRAMU R5,R5SAVEU
00826 3415F F002A  3873              ALDRAM R5,R5SAVE                  ;SAVE R/W FLAGS FOR LATER
00827 2E140 F0023  3874              SUBB DQ,FOUTL,MSK1,ZERO & NOP D#35 ;IS DISPLACEMENT (L) = OR < 35?
00828 37160 3587F  3875              MOVE SMCST,ZERO & CJMP NALB,ECCOR14 ;IF YES, ERROR IS IN ECC FIELD
                3876    ;
00829 373F2 F02CD  3877              RDRAMB R2,RECLNTH                 ;GET RECORD LENGTH...
```

Addr Line DISK ECC ERROR PROCESSING

```
0082A 37351 F0024  3878            RDMSKB R1,D#36                  ;TIMES 36 (BITS PER WORD) =...
0082B 323F4 1F0CF  3879            QTOR R4 & JSR MULTPLYA          ;BITS PER RECORD (R0)
0082C 05340 F0023  3880            ADD DB,B,MSK1,R0 & NOP D#35     ;R0 + 35 = MAXLNTH
                   3881    ;
0082D 3710F F00A6  3882            RESETRAM PARCNT                 ;RESET PARITY ERROR RETRY COUNT
0082E C1073 5F02E  3883            RD HSSM,RDECCPAT,R3 & LDAR $    ;READ ERROR PATTERN...
0082F C1073 FFFFF  3884            RD HSSM,RDECCPAT,R3             ;TWICE AND...
                   3885            JSR.P XMCTPAR                   ;CHECK CT-BUS PARITY
00830 E0403 1F35B  3885 +          SETPGE XMCTPAR_SHR_H#B & JSRP XMCTPAR
                   3886            BNZ.P HSSMPAR                   ;IF ERROR, PROCESS IT
00831 E0403 33B62  3886 +          SETPGE HSSMPAR_SHR_H#B & BNZP HSSMPAR
00832 3413F F00A7  3887            ALDRAM R3,ECCPAT                ;SAVE ERROR PATTERN FOR LATER
00833 45053 F00FF  3888            TSTRBIT R3,H#FF                 ;ISOLATE PATTERN LSB'S...
00834 321FF F001E  3889            QLDRAM SEN522                   ;AND SAVE THEM FOR SENSE
00835 60331 4F001  3890            XOR AQ,B,R3,R1 & PSHLD 1        ;ISOLATE THE MSB'S AND...
                   3891            LSHFT2 R1                       ;MOVE THEM INTO POSITION
00836 01711 887FF  3891 +          ADD AB,LSBZ,R1,R1 & CLPCNT EVCNTZ
00837 3491F F001D  3892            ALDRAMU R1,SEN521               ;SAVE THEM FOR SENSE
                   3893    ;
00838 29040 FFFFF  3894            SUBB AB,Q,R4,R0                 ;IS L (R4) > MAXLNTH (R0)?
00839 FFFFF 3408E  3895            CJMP AGB,ECCOR23                ;IF YES, ERROR MAY BE OUTSIDE...
                   3896    ;
0083A 1D344 F0023  3897    ECCOR05A: SUBA DB,B,MSK1,R4 & NOP D#35  ;SUBTRACT ECC FIELD FROM DISPLACEMENT
0083B 373F0 F02CD  3898            RDRAMB R0,RECLNTH               ;GET RECORD LENGTH...
                   3899            JSR.P FOUR5DN                   ; X 4.5 = BYTELNTH (R0) AND...
0083C E0400 1F368  3899 +          SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
0083D 37353 F0001  3900            RDMSKB R3,1                     ;PRESET BYTE POINTER = 1 (R3)
                   3901    ;
0083E 1D344 F0008  3902    ECCOR06:  SUBA DB,B,MSK1,R4 & NOP H#8     ;L -8 = L
0083F 333F4 31842  3903            OR ZB,B,,R4 & CJMP NCARRY,ECCOR07 ;IF L IS NEGATIVE, JUMP
00840 FFFFF 33042  3904            BOZ ECCOR07                     ;IF L IS 0, JUMP
00841 0B3F3 E083E  3905            INCRB R3 & JMP ECCOR06          ;INCR. BYTE POINTER & JUMP BACK
                   3906    ;
00842 2B3F4 FFFFF  3907    ECCOR07:  SUBB ZB,B,,R4                   ;2'S COMP. OF L = SHIFT COUNT (S)
00843 29303 FFFFF  3908            SUBB AB,B,R0,R3                 ;BYTELNTH - R3 = RELATIVE BYTE (B)
00844 4505F F0004  3909    ECCOR07A: TSTRBIT R15,4                 ;U-INHIBIT 2?
00845 433F0 33851  3910            RESETR R0 & BNZ ECCOR08         ;IF SET, JUMP
00846 370FF F0025  3911            RDRAMQ SEN216                   ;GET SENSE 16, FORMAT 2
00847 6605F F0001  3912            TSTQMSK 1                       ;VERIFY OPERATION?
00848 FFFFF 33051  3913            BOZ ECCOR08                     ;YES, JUMP
                   3914    ;
00849 373F2 F02CD  3915            RDRAMB R2,RECLNTH               ;GET RECORD LENGTH
0084A 373F1 F0029  3916            RDRAMB R1,RELREC                ;GET RELATIVE RECORD NO.
0084B 4505E F2040  3917            TSTRBIT R14,2040                ;READ FIRST OR READ LAST FLAGS?
0084C 333F1 3384F  3918            OR ZB,B,,R1 & BNZ $+3           ;IF EITHER, JUMP
0084D 133F1 3384F  3919            DECRB R1 & BNZ $+2              ;DECR. REL. REC. #; NON-0, JUMP
0084E 273F1 F02D0  3920            SAB DZ,B,RAME,R1 & NOP MAXREL   ;MAX. RELATIVE RECORD NO.-1
                   3921    ;
0084F 433F0 138CF  3922            RESETR R0 & CJSR NALUZ,MULTPLYA ;REC. LNTH.X RELATIVE REC...
                   3923            JSR.P FOUR5DN                   ;TIMES 4.5 = RECORD LOC. (R0)
00850 E0400 1F368  3923 +          SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
```

Addr Line DISK ECC ERROR PROCESSING

```
00851 01330 5F044  3924    ECCOR08:  ADD AB,B,R3,R0 & LDAR ECCOR07A     ;RECORD LOC.+ B = ERROR LOC.
00852 34302 4F001  3925              RTOR R0,R2 & PSHLD 1               ;ERROR LOC. TO R2 ALSO
                   3926              LSHFT2 R0                          ;MOVE MSB'S INTO POSITION...
00853 01700 887FF  3926 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
00854 A210F FFFFF  3927              WRTU HSSMA,LDMDARM,R0              ;AND LOAD THEM
00855 3705F F00FF  3928              RDMSKQ H#FF
00856 40320 FFFFF  3929              AND AQ,B,R2,R0                     ;ADDR. 8 LSB'S TO R0...
00857 9200F FFFFF  3930              WRTL HSSMA,LDMDARL,R0              ;AND LOAD THEM
00858 3705F F0003  3931              RDMSKQ 3                           ;PRESET BYTE COUNT = 3
00859 433F1 FFFFF  3932              RESETR R1
                   3933    ;
0085A C10D0 FFFFF  3934    ECCOR09:  RD HSSM,RDMDBDA,R0                 ;READ DATA BYTE
0085B C10D0 FFFFF  3935              RD HSSM,RDMDBDA,R0
0085C 120FF C0003  3936              DECRQ & LDCT 3                     ;DECR. BYTE COUNT
0085D 31301 33060  3937              OR AB,B,R0,R1 & BOZ ECCOR10        ;IF 0, JUMP; MERGE PREVIOUS DATA
0085E 01711 9805E  3938              ADD AB,LSBZ,R1,R1 & CJPCNT EVCNTZ,$ ;SHIFT DATA 8 PLACES
0085F 083F2 E085A  3939              INCRB R2 & JMP ECCOR09             ;INCR. ERROR LOC. & GET NEXT BYTE
                   3940    ;
                   3941    ECCOR10:  JSR.P XMCTPAR                      ;CHECK CT-BUS PARITY
00860 E0403 1F35B  3941 +            SETPGE XMCTPAR_SHR_H#B & JSRP XMCTPAR
                   3942              BNZ.P HSSMPAR                      ;IF AN ERROR, PROCESS IT
00861 E0403 33B62  3942 +            SETPGE HSSMPAR_SHR_H#B & BNZP HSSMPAR
00862 370FF F00A7  3943              RDRAMQ ECCPAT                      ;GET THE ERROR PATTERN AND...
00863 D0FF5 FFFFF  3944              RDXFR SMC,,R5                      ;MOVE IT TO BITS 12-23
00864 333F4 FFFFF  3945              OR ZB,B,,R4
00865 FFFFF 33068  3946              BOZ ECCOR11                        ;IF SHIFT COUNT (S) = 0, CONTINUE
00866 335F5 FFFFF  3947              OR ZB,RSBZ,,R5                     ;RIGHT SHIFT ERROR PATTERN
00867 133F4 E0865  3948              DECRB R4 & JMP $-2                 ;DECR. SHIFT COUNT & CHECK AGAIN
                   3949    ;
00868 34324 FFFFF  3950    ECCOR11:  RTOR R2,R4                         ;ERROR LOC. TO R4
                   3955    ;
00869 61351 1F097  3956    ECCOR12:  XOR AB,B,R5,R1 & JSR MASKR4        ;CORRECT BITS IN ERROR INTO R1
0086A 9200F 4F001  3957              WRTL HSSMA,LDMDARL,R0 & PSHLD 1    ;LOAD ERROR LOC. LSB'S,
                   3958              LSHFT2 R4                          ;MOVE MSB'S INTO POSITION AND...
0086B 01744 887FF  3958 +            ADD AB,LSBZ,R4,R4 & CLPCNT EVCNTZ
0086C A214F 1F096  3959              WRTU HSSMA,LDMDARM,R4 & JSR MASKR1 ;LOAD THEM
0086D 9230F 4F003  3960              WRTL HSSMA,LDMDAT,R0 & PSHLD 3     ;WRITE CORRECTED BITS 0-7
0086E 336F1 887FF  3961              LSHFTDZ R1                         ;LEFT SHIFT CORRECTED DATA 4 PLACES
                   3962    ;
0086F 133F2 FFFFF  3963              DECRB R2                           ;DECR. ERROR LOC.
00870 34324 1F097  3964              RTOR R2,R4 & JSR MASKR4            ;INTO R4 ALSO
00871 9200F 4F001  3965              WRTL HSSMA,LDMDARL,R0 & PSHLD 1    ;LOAD ERROR LOC. LSB'S,
                   3966              LSHFT2 R4                          ;MOVE MSB'S INTO POSITION AND...
00872 01744 887FF  3966 +            ADD AB,LSBZ,R4,R4 & CLPCNT EVCNTZ
00873 A214F FFFFF  3967              WRTU HSSMA,LDMDARM,R4              ;LOAD THEM
00874 37350 F00FF  3968              RDMSKB R0,H#FF
00875 D0FF0 FFFFF  3969              RDXFR SMC,,R0
```

Addr Line DISK ECC ERROR PROCESSING

```
00876 41310 FFFFF  3970            AND AB,B,R1,R0                    ;ISOLATE BITS 8-15 ONLY
00877 A230F 4F003  3971            WRTU HSSMA,LDMDAT,R0 & PSHLD 3    ;WRITE CORRECTED BITS 8-15
00878 336F1 887FF  3972            LSHFTDZ R1                        ;LEFT SHIFT CORRECTED DATA 4 PLACES
                   3973    ;
00879 14324 FFFFF  3974            SBA AZ,B,R2,R4                    ;DECR. ERROR LOC. INTO R4
0087A 323F1 1F097  3975            QTOR R1 & JSR MASKR4              ;CORRECTED DATA 16-23 INTO R1
0087B 9200F 4F001  3976            WRTL HSSMA,LDMDARL,R0 & PSHLD 1   ;LOAD ERROR LOC. LSB'S,
                   3977            LSHFT2 R4                         ;MOVE MSB'S INTO POSITION AND...
0087C 01744 887FF  3977 +          ADD AB,LSBZ,R4,R4 & CLPCNT EVCNTZ
0087D A214F 1F096  3978            WRTU HSSMA,LDMDARM,R4 & JSR MASKR1 ;LOAD THEM
0087E 9230F FFFFF  3979            WRTL HSSMA,LDMDAT,R0              ;WRITE CORRECTED BITS 16-23
                   3980    ;
0087F 37352 F0093  3981    ECCOR14: RDMSKB R2,93                      ;LOC. OF ECC LOG
00880 FFFFF 1F099  3982            JSR ECCLOG                        ;LOG CORRECTABLE ECC
00881 E1011 1F0C0  3983            CTL HSSM,RSECTLR & JSR CORLOG     ;INCR. CORRECTABLE ERROR COUNT
00882 373F5 F00A5  3984            RDRAMB R5,R5SAVEU                 ;RESTORE READ/WRITE FLAGS
00883 D0FF5 FFFFF  3985            RDXFR SMC,,R5
00884 353F5 F002A  3986            OR DB,B,RAME,R5 & NOP R5SAVE
                   3987            JSR.P RDSMADDR                    ;COMPUTE NEXT REC. SM-ADDRESS
00885 E0400 1F4C8  3987 +          SETPGE RDSMADDR_SHR_H#B & JSRP RDSMADDR
00886 4505F F0020  3992            TSTRBIT R15,20                    ;NO ECC CORRECTION FLAG?
00887 343B3 3388C  3993            RTOR R11,R3 & BNZ ECCOR15         ;YES, JUMP; CYL. TO R3
00888 45055 F0020  3994            TSTRBIT R5,20                     ;SM-COMPLETE?
                   3995            CJSR.P ALUZ,WRT050                ;NO, RELOAD THE HEADER INFO.
00889 E0400 135BA  3995 +          SETPGE WRT050_SHR_H#B & CJSRP ALUZ,WRT050
0088A 5535E F4000  3996            RSTRBIT R14,4000                  ;RESET READ/WRITE STARTED FLAG...
0088B FFFFF E0395  3997            JMP IRPT01A                       ;AND READ THE NEXT RECORD
                   3998    ;
0088C 37350 F0053  3999    ECCOR15: RDMSKB R0,53                      ;SENSE 7 = DATA ECC CORRECTABLE
0088D FFFFF E069C  4000            JMP SMERR33                       ;SET ECC CORRECTABLE FLAG
                   4030    ;
0088E 373F5 F002A  4031    ECCOR23: RDRAMB R5,R5SAVE                  ;RESTORE SAVED R5
0088F 2E340 F000C  4032            SUBB DQ,B,MSK1,R0 & NOP D#12      ;IF L IS 12 OR MORE BITS BEFORE...
00890 433F1 34813  4033            RESETR R1 & CJMP NAGB,ECCOR02     ;THE DATA FIELD, ECC UNCORRECTABLE
00891 133F0 FFFFF  4034            DECRB R0                          ;DECR. R0
00892 33FF1 33891  4035            OR ZB,LSB1,,R1 & BNZ $-1          ;SHIFT 1 INTO MASK (R1)
00893 51113 FFFFF  4036            MASK AB,FOUTL,R1,R3               ;IF THE ERROR PATTERN BITS ARE...
00894 FFFFF 33813  4037            BNZ ECCOR02                       ;OUTSIDE DATA, ECC UNCORRECTABLE
00895 E1011 E083A  4038            CTL HSSM,RSECTLR & JMP ECCOR05A   ;ELSE, CORRECT THE DATA
                   4039    ;
00896 34310 E0898  4040    MASKR1:  RTOR R1,R0 & JMP $+2              ;MOVE R1 TO R0 & JUMP
00897 34340 FFFFF  4041    MASKR4:  RTOR R4,R0                        ;MOVE R4 TO R0
00898 45340 AF0FF  4042            AND DB,B,MSK1,R0 & RTS H#FF       ;ISOLATE 8 LSB'S & RETURN
```

Addr Line DISK ECC ERROR PROCESSING

```
                    4044
                    4045    ;     THE ECC LOGGING ROUTINE GENERATES E.C.C. HISTORY DISPLAYS TO AID IN
                    4046    ;LOCATING POSSIBLE BAD AREAS OF PACKS.
                    4047    ;     THE DISPLAYS ARE:
                    4048
                    4049    ; DISPLAY       MEANING
                    4050    ;   431 CORRECTABLE E.C.C. HISTORY 1
                    4051    ;   432 CORRECTABLE E.C.C. HISTORY 2
                    4052    ;   433 CORRECTABLE E.C.C. COUNT (ALL DRIVES)
                    4053    ;   434 UNCORRECTABLE E.C.C. HISTORY 1
                    4054    ;   435 UNCORRECTABLE E.C.C. HISTORY 2
                    4055    ;   436 UNCORRECTABLE E.C.C. COUNT (ALL DRIVES)
                    4056    ;
                    4057    ;     THE ABOVE DISPLAYS MAY BE CLEARED AT ANY TIME BY
                    4058    ;RUNNING ILD 8.
                    4059    ;
                    4060    ;     THE FORMAT OF THE E.C.C. HISTORY DATA IS:
                    4061    ;
                    4062    ;35,34 - - - - - - - -24,23 - - -19,18 - - - -12,11 - - - - -06,05 - - - 00
                    4063    ; 0    C Y L I N D E R    HEAD NO.   RECORD NO.   ERROR COUNT   PHYS. DRIVE
                    4064
                    4065    ;     UPON OCCURRENCE OF AN E.C.C. ERROR THE HISTORY DISPLAY WITH THE LOWEST
                    4066    ;ERROR COUNT WILL BE UPDATED.  IF THE COUNTS OF BOTH DISPLAYS ARE EQUAL, THEN
                    4067    ;HISTORY 1 WILL BE UPDATED.
                    4068
                    4069    ;UPON ENTRY:  R2  = LOC. OF HISTORY WORD
                    4070    ;             R9  = CURRENT RECORD
                    4071    ;             R11 = CURRENT CYLINDER
                    4072    ;             R12 = CURRENT HEAD
                    4073    ;             R13 = PHYSICAL DRIVE NUMBER
                    4074    ;REGISTERS USED:  R0-R4,Q & IREG
                    4075
                    4076
00899 5535E F0004   4077    ECCLOG:   RSTRBIT R14,4                     ;RESET HISTORY 2 FLAG
0089A 05042 F0008   4078              ADD DB,Q,MSK1,R2 & NOP D#8        ;LOC. OF ECC COUNT
0089B 321FE FFFFF   4079              QLDREG IREG
                    4080              JSR.P INCCNT                      ;INCREMENT THE COUNT
0089C E0400 1F120   4080 +            SETPGE INCCNT_SHR_H#B & JSRP INCCNT
                    4081    ;
0089D 3412E 1F0CD   4082    ECCLOG1:  ALDREG R2,IREG & JSR SMBRDA       ;GET HISTORY LOG
0089E 323F1 FFFFF   4083              QTOR R1                           ;MOVE HISTORY CYL. TO R1
0089F 4535D F00FF   4084              MSKRBIT R13,H#FF                  ;DELETE ANY STATUS
008A0 3490E FFFFF   4085              ALDREGU R0,IREG                   ;HISTORY HEAD/RECORD TO IREG
008A1 347C4 4F002   4086              OR AZ,LSBZ,R12,R4 & PSHLD 2       ;CUR. HEAD TO R4 AND...
                    4087              LSHFT2 R4                         ;MOVE IT INTO POSITION
008A2 01744 887FF   4087 +            ADD AB,LSBZ,R4,R4 & CLPCNT EVCNTZ
008A3 31049 FFFFF   4088              OR AB,Q,R4,R9                     ;MERGE CURRENT RECORD TO Q
008A4 661E0 FFFFF   4089              XOR DQ,FOUTL,IREG,ZERO            ;CHECK HEAD/RECORD MATCH
008A5 321FE 338B1   4090              QLDREG IREG & BNZ ECCLOG3         ;IF NOT, JUMP; HEAD/REC. TO IREG
                    4091    ;
008A6 45050 F003F   4092              TSTRBIT R0,3F                     ;ISOLATE HISTORY DRIVE NO.
008A7 601D0 FFFFF   4093              XOR AQ,FOUTL,R13,ZERO             ;DRIVE MATCH?
```

```
 Addr            Line DISK ECC ERROR PROCESSING

008A8 610B1 338B1  4094            XOR AB,Q,R11,R1 & BNZ ECCLOG3     ;NO, JUMP; CHECK CYL. MATCH
008A9 FFFFF 338B1  4095            BNZ ECCLOG3                       ;IF NO MATCH, JUMP
008AA 45050 F0FC0  4096            TSTRBIT R0,H#FC0                  ;ISOLATE THE ERROR COUNT
008AB 66150 F0FC0  4097            TSTQEQU H#FC0                     ;CHECK FOR MAX. COUNT
008AC 3412E 338AE  4098            ALDREG R2,IREG & BNZ ECCLOG2      ;IF NOT MAX., JUMP
008AD 60300 E08AF  4099            XOR AQ,B,R0,R0 & JMP $+2          ;RESET ERROR COUNT
008AE 05340 F0040  4100   ECCLOG2:  ADD DB,B,MSK1,R0 & NOP H#40        ;INCR. ERROR COUNT
008AF 5535E F0004  4101            RSTRBIT R14,4                     ;RESET HISTORY 2 FLAG
008B0 3401F E08CE  4102            RTOQ R1 & JMP SMBWRTA             ;SAVE NEW COUNT & RETURN
                   4103   ;
008B1 4505E F0004  4104   ECCLOG3:  TSTRBIT R14,4                     ;CHECK HISTORY 2 FLAG
008B2 3400F 338B7  4105            RTOQ R0 & BNZ ECCLOG4             ;IF SET, JUMP
                   4106   ;
008B3 3535E F0004  4107            SETRBIT R14,4                     ;SET HISTORY 2 FLAG
008B4 46353 F0FC0  4108            AND DQ,B,MSK2,R3 & NOP H#FC0      ;ISOLATE HISTORY 1 COUNT IN R3
008B5 05352 F0003  4109            ADD DB,B,MSK2,R2 & NOP D#3        ;LOC. OF HISTORY 2 LOG
008B6 FFFFF E089D  4110            JMP ECCLOG1                       ;GET & CHECK HISTORY 2
                   4111   ;
008B7 45050 F0FC0  4112   ECCLOG4:  TSTRBIT R0,H#FC0                  ;ISOLATE HISTORY 2 COUNT IN Q
008B8 28130 FFFFF  4113            SUBB AQ,FOUTL,R3,ZERO             ;IF HISTORY 1 COUNT...
008B9 343B1 340BB  4114            RTOR R11,R1 & CJMP AGB,$+2        ; > HISTORY 2, UPDATE HISTORY 2
008BA 1D342 F0003  4115            SUBA DB,B,MSK1,R2 & NOP D#3       ;LOC. OF HISTORY 1
                   4116   ;
008BB 373E0 FFFFF  4117            RDREGB IREG,R0                    ;GET CURRENT HEAD/RECORD
008BC D0FF0 FFFFF  4118            RDXFR SMC,,R0                     ;STROBE XFER. REG.
008BD 313D0 FFFFF  4119            OR AB,B,R13,R0                    ;MERGE THE DRIVE NO.
008BE 3412E E08AE  4120            ALDREG R2,IREG & JMP ECCLOG2      ;LOG NEW E.C.C. ERROR
                   4121
                   4122
008BF 0F303 E08C1  4123   UNCORLOG: ADD DZC,B,ZERO,R3 & JMP $+2       ;INCREMENT FOR UNCORRECTABLE
008C0 37753 F8000  4124   CORLOG:   OR DZ,LSBZ,MSK2,R3 & NOP H#8000   ;INCREMENT FOR CORRECTABLE
                   4125              JSR.P ERRLOG
008C1 E0400 1F350  4125 +          SETPGE ERRLOG_SHR_H#B & JSRP ERRLOG
008C2 05342 F0003  4126            ADD DB,B,MSK1,R2 & NOP D#3        ;LOC. OF ECC ERROR LOG
008C3 3412E 1F0CD  4127            ALDREG R2,IREG & JSR SMBRDA       ;GET THE LOG AND...
008C4 01330 FFFFF  4128            ADD AB,B,R3,R0                    ;INCREMENT IT
008C5 FFFFF 318C7  4129            CJMP NCARRY,$+2                   ;IF NO CARRY, JUMP
008C6 0A0FF FFFFF  4130            INCRQ                             ;INCR. CORRECTABLE MSB'S
008C7 3412E 1F0CE  4131            ALDREG R2,IREG & JSR SMBWRTA      ;SAVE INCREMENTED LOG
008C8 45150 F0040  4132            TSTBIT R0,40                      ;MAX. UNCORRECTABLE ERRORS?
                   4133              BNZ.P LOGOFL                    ;IF YES, LOG OVERFLOW
008C9 E0404 33825  4133 +          SETPGE LOGOFL_SHR_H#B & BNZP LOGOFL
008CA 4605F F0002  4134            TSTQBIT 2                         ;MAX. CORRECTABLE ERRORS?
008CB FFFFF A37FF  4135            CRTS ALUZ                         ;IF NOT, RETURN
008CC FFFFF E2025  4136            JMP LOGOFL                        ;ELSE, SET LOG OVERFLOW
                   4137   ;
                   4138
                   4140   TITLE CDC O.S. - BOOKKEEPING
                   4148
008CD FFFFF E003B  4149   SMBRDA:   JMP SMBRD
```

Addr Line CDC O.S. -- BOOKKEEPING

```
008CE FFFFF E0044  4150    SMBWRTA:  JMP SMBWRT
008CF FFFFF E02C0  4151    MULTPLYA: JMP MULTPLY
```

Addr Line CDC O.S. - BOOKKEEPING

```
                    4153
                    4154
                    4155    ;    THE BOOKKEEPING ROUTINES ARE DIVIDED INTO FOUR MAJOR SECTIONS:
                    4156    ;THE FIRST SECTION KEEPS TRACK OF DISK (SM) TRANSFER LENGTH, AMPERIF CYLINDER,
                    4157    ;HEAD & RECORD NUMBERS.  IT SETS THE ADVANCE HEAD OR ADVANCE CYLINDER FLAGS
                    4158    ;WHEN THE DATA TRANSFER IS ABOUT TO CROSS BEYOND ONE TRACK.  IT KEEPS TRACK
                    4159    ;OF THE DATA BUFFER RELATIVE RECORD NUMBER.  IT ALSO CHECKS FOR THE TRANSFER
                    4160    ;EXCEEDING THE PACK LIMITS.
                    4161    ;
                    4162    ;THE SECOND SECTION IS EXECUTED IMMEDIATELY AFTER THE FIRST AND IT KEEPS TRACK
                    4163    ;OF THE UNIVAC CYLINDER, HEAD AND RECORD NUMBERS.
                    4164    ;
                    4165    ;THE THIRD SECTION IS EXECUTED INDEPENDENTLY OF THE FIRST TWO AND IT KEEPS
                    4166    ;TRACK OF THE CHANNEL (U) TRANSFER LENGTH.  IT COMPUTES THE TRANSFER LENGTH
                    4167    ;OF THE NEXT CHANNEL (U) OPERATION AND THE RELATIVE WORD NUMBER.  IT
                    4168    ;DETERMINES WHEN THE CHANNEL (U) OPERATION IS COMPLETE.
                    4169    ;
                    4170    ;;    THE FOURTH SECTION IS EXECUTED AFTER THE SECOND.  IT PRE-DETERMINES
                    4171    ;WHAT OPERATION IS TO BE PERFORMED NEXT.  A MPC ROUTINE JUMP LOCATION IS
                    4172    ;LOADED (BUT NOT EXECUTED) INTO THE XMCI BOARD.  AN XMC NEXT OPERATION JUMP
                    4173    ;LOCATION IS LOADED INTO AN SMB MEMORY LOCATION (ADVLOC) TO BE EXECUTED AFTER
                    4174    ;THE SM-DONE INTERRUPT IS RECEIVED.  THIS JUMP IS DONE IN THE ROUTINE 'ADVREC'.
                    4175    ;THIS FOURTH SECTION IS SMBOOK20-SMBOOK31.
                    4176    ;
008D0 4505F F0400   4177    SMBOOK:   TSTRBIT R15,400                   ;SM-ERROR FLAG?
                    4178              BNZ.P SMBOOK27                    ;YES, JUST RELOAD NEXT RECORD
008D1 E0403 33AF7   4178 +            SETPGE SMBOOK27_SHR_H#B & BNZP SMBOOK27
                    4179              JSR.P SEEKSENS                    ;SAVE CYL./HEAD FOR SENSE
008D2 E0400 1F1DA   4179 +            SETPGE SEEKSENS_SHR_H#B & JSRP SEEKSENS
008D3 3419F F0015   4180              ALDRAM R9,SEN412                  ;SAVE REC. FOR SENSE 12, FORMAT 4,5
008D4 3710F F0018   4181              RESETRAM SEN415                   ;RESET SM-RETRY COUNT
008D5 343B1 FFFFF   4182              RTOR R11,R1                       ;CUR. CYLINDER TO R1
008D6 45055 F1000   4183              TSTRBIT R5,1000                   ;FORMAT ACTIVE FLAG?
008D7 343C2 338FC   4184              RTOR R12,R2 & BNZ SMBOOK02        ;IF YES, JUMP; HEAD TO R2
                    4185    ;
008D8 373F0 F02CD   4186              RDRAMB R0,RECLNTH                 ;GET RECORD LENGTH
008D9 4505E F0008   4187              TSTRBIT R14,8                     ;HOME ADDRESS FLAG?
008DA 34394 330DF   4188              RTOR R9,R4 & BOZ SMBOOK00         ;IF NOT, CONTINUE; RECORD TO R4
008DB 05340 F0002   4189              ADD DB,B,MSK1,R0 & NOP D#2        ;ADD 2 MORE WORDS FOR COUNT AREA
008DC 4505E F4000   4190              TSTRBIT R14,4000                  ;READ/WRITE STARTED?
008DD FFFFF 338DF   4191              BNZ SMBOOK00                      ;IF YES, CONTINUE
008DE 05340 F0002   4192              ADD DB,B,MSK1,R0 & NOP D#2        ;ADD 2 WORDS FOR HOME ADDRESS
                    4193    ;
008DF 1D0F0 F002B   4194    SMBOOK00: SUBA DB,Q,RAME,R0 & NOP STRTWD    ;REC. LENGTH - START WORD = Q
008E0 28377 FFFFF   4195              SUBB AQ,B,R7,R7                   ;XFER. LNTH.- Q = XFER. LNTH.
008E1 333F7 318E4   4196              OR ZB,B,,R7 & CJNP NCARRY,SMBOOK0A ;IF NEGATIVE NO., JUMP
008E2 FFFFF 330E4   4197              BOZ SMBOOK0A                      ;IF Q, JUMP
008E3 FFFFF E08E6   4199              JNP SMBOOK0B                      ;$IF NON-ZERO, CONTINUE
                    4208    ;
```

Addr Line CDC O.S. - BOOKKEEPING

```
                4209    ;   SM-complete
                4210    ;
008E4 35355 F0020  4211    SMBOOK0A: SETRBIT R5,20                    ;SET SM-COMPLETE FLAG
008E5 433F7 FFFFF  4212              RESETR R7                        ;RESET SM-XFER. LENGTH
                4213    ;
008E6 3710F F002B  4214    SMBOOK0B: RESETRAM STRTWD                  ;NEXT REC. STARTS AT WORD 0
008E7 2D0F4 F02D6  4215              SUBB DB,Q,RAME,R4 & NOP AMPREC   ;CHECK FOR MAX. RECORD
008E8 0B3F4 348FC  4216              INCRB R4 & CJMP NAGB,SMBOOK02    ;IF YES, JUMP; INCR. RECORD
                4217    ;
008E9 3414F F0028  4222    SMBOOK01: ALDRAM R4,NEXTREC                ;SAVE NEXT RECORD NO.
008EA 3412F F0027  4224              ALDRAM R2,NEXTHD                 ;SAVE NEXT HEAD NO.
008EB 3411F F0026  4225              ALDRAM R1,NEXTCYL                ;SAVE NEXT CYLINDER
008EC 45055 F1000  4226              TSTRBIT R5,1000                  ;FORMAT ACTIVE?
008ED FFFFF A3FFF  4227              CRTS NALUZ                       ;IF YES, RETURN NOW
                4228    ;
                4229              INCSAB RELREC                    ;Incr. Relative Record no.
008EE 0F3F0 F0029  4229 +            ADD      DZC,B,RAME,R0 & NOP RELREC
008EF 3410F F0029  4229 +            ALDRAM   R0,RELREC
008F0 4505E F0008  4230              TSTRBIT R14,8                    ;HOME ADDRESS FLAG?
008F1 0C301 3311A  4231              ADD AZC,B,R0,R1 & BOZ SMBOOK10   ;IF NOT, UNIVAC BOOKKEEPING
                4232    ;
008F2 3415E FFFFF  4233              ALDREG R5,IREG                   ;FLAGS TO IREG
008F3 0F3F2 F02CD  4234              ADD DZC,B,RAME,R2 & NOP RECLNTH  ;GET RECORD LENGTH + 1
008F4 0B3F2 1F0CF  4235              INCRB R2 & JSR MULTPLYA          ;+ 1 MORE & CONVERT WORD NO.
008F5 05340 F0004  4236              ADD DB,B,MSK1,R0 & NOP D#4       ;+ 4 MORE FOR H.A. & COUNT
                4237              JSR.P FOUR5DN                    ;CONVERT TO BYTES
008F6 E0400 1F368  4237 +            SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
008F7 34302 FFFFF  4238              RTOR R0,R2                       ;NEXT DISK ADDR. TO R2
                4239              JSR.P CHKBUSY                    ;CHECK FOR HSSM BUSY
008F8 E0400 1F535  4239 +            SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
008F9 9102F FFFFF  4240              WRTL HSSM,LDSARL,R2              ;LOAD NEXT DISK ADDR. LOWER AND...
008FA A111F FFFFF  4241              WRTU HSSM,LDSARM,R1              ;NEXT DISK. ADDR. UPPER
008FB 373E5 E091A  4242              RDREGB IREG,R5 & JMP SMBOOK10    ;RESTORE FLAGS & JUMP
                4243    ;
                4244    SMBOOK02: EQU $
008FC 37354 F0001  4246              RDHSKB   R4,1                    ;$Set record no.= 1
008FD 4505E F0080  4252              TSTRBIT R14,80                   ;< 1-TRACK FLAG?
008FE FFFFF 33101  4253              BOZ $+3                          ;<IF NOT, JUMP
008FF 45055 F0020  4254              TSTRBIT R5,20                    ;<SM-COMPLETE FLAG?
00900 FFFFF 33113  4255              BOZ SMBOOK05                     ;<IF NOT, SET END-OF-PACK
00901 2D0F2 F02D5  4257              SUBB DB,Q,RAME,R2 & NOP MAXHDA   ;CHECK FOR MAX. HEAD
00902 FFFFF 34908  4258              CJMP NAGB,SMBOOK04               ;IF YES, JUMP
                4259    ;
                4260    ;   Advance Head
```

Addr Line CDC O.S. - BOOKKEEPING

```
                4261    ;
00903 3535F F0800 4262          SETRBIT R15,800                 ;SET ADV. HEAD FLAG
00904 45055 F1000 4263          TSTRBIT R5,1000                 ;FORMAT ACTIVE FLAG?
00905 0B3F2 330E9 4264          INCRB R2 & BOZ SMBOOK01         ;INCR. HEAD NO.; IF NOT, JUMP
00906 3419F F0028 4265          ALDRAM R9,NEXTREC               ;SAVE CURRENT RECORD
00907 FFFFF E08EA 4269          JMP SMBOOK01+1
                4271    ;
                4272    ;   Advance Cylinder
                4273    ;
00908 433F2 FFFFF 4274  SMBOOK04: RESETR R2                     ;RESET HEAD NO.
00909 3535F F1000 4275          SETRBIT R15,1000                ;SET ADVANCE CYLINDER FLAG
0090A 2D1F1 F02D3 4276          SUBB DB,FOUTL,RAME,R1 & NOP MAXCYLA ;IS CUR. CYL.< MAX?
0090B 0B3F1 3410F 4278          INCRB R1 & CJMP AGB,SMBOOK4A    ;<IF YES, JUMP; INCR. CYL.+ 1
0090C 45055 F0020 4286          TSTRBIT R5,20                   ;SM-COMPLETE FLAG SET?
0090D FFFFF 338E9 4288          BNZ     SMBOOK01                ;$Yes, continue
0090E FFFFF E0913 4297          JMP SMBOOK05                    ;<End-of-pack
                4298    ;
0090F 45055 F0800 4299  SMBOOK4A: TSTRBIT R5,800                ;<WRITE OPERATION?
00910 FFFFF 330E9 4300          BOZ SMBOOK01                    ;<NO, CONTINUE
00911 4505F F0200 4304          TSTRBIT R15,200                 ;<FORMAT FLAG SET?
00912 FFFFF 330E9 4305          BOZ SMBOOK01                    ;<NO, CONTINUE
                4306    ;
                4313    ;
00913 45055 F0800 4314  SMBOOK05: TSTRBIT R5,800                ;WRITE OPERATION?
00914 FFFFF 33117 4315          BOZ SMBOOK5A                    ;NO, SET END-OF-PACK
00915 4505F F0200 4319          TSTRBIT R15,200                 ;FORMAT FLAG SET?
00916 FFFFF 33918 4320          BNZ $+2                         ;YES, DON'T SET END-OF-PACK
00917 35355 F0040 4321  SMBOOK5A: SETRBIT R5,40                 ;SET END-OF-PACK FLAG
00918 3535E F0400 4322          SETRBIT R14,400                 ;SET U-COMPLETE FLAG
00919 433F4 E08E4 4323          RESETR R4 & JMP SMBOOK0A        ;RESET NEXT RECORD & JUMP
                4324    ;
0091A 45055 F0800 4326  SMBOOK10: TSTRBIT R5,800                ;<WRITE OPERATION?
0091B FFFFF 33920 4338          BNZ SMBOOK12                    ;IF IT IS, JUMP
0091C 4505F F0200 4339          TSTRBIT R15,200                 ;NO DATA XFER. FLAG?
0091D FFFFF 3312B 4340          BOZ SMBOK12A                    ;IF NOT, JUMP
0091E 3535E F0400 4341          SETRBIT R14,400                 ;SET U-COMPLETE FLAG
```

```
 Addr            Line CDC O.S. - BOOKKEEPING

0091F FFFFF E092B  4342            JMP SMBOK12A                    ;COMPUTE NEXT RECORD JUMP
                   4343    ;
                   4344    SMBOOK12: EQU $
00920 45055 F0008  4350            TSTRBIT R5,8                    ;U-SIDE WAITING?
00921 60355 33123  4351            XOR AQ,B,R5,R5 & BOZ $+2        ;NO, JUMP; YES, RESET IT...
00922 35355 F0200  4352            SETRBIT R5,200                  ;AND SET RESTART U-SIDE FLAG
00923 373F0 F02CD  4353            RDRAMB R0,RECLNTH               ;GET THE RECORD LENGTH AND...
                   4354            JSR.P FOUR5DN                   ;CONVERT IT TO BYTES
00924 E0400 1F368  4354 +          SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
00925 3400F FFFFF  4355            RTOQ R0                         ;MOVE IT TO Q
                   4356            LDWD R1,RBAU                    ;Get the Record Byte Address
00926 373F1 F00AC  4356 +          RDRAMB   R1,RBAU               ;GET 12 MSBS
00927 D0FF1 FFFFF  4356 +          RDXFR    SMC,,R1                ;PUT IN UPPER REGISTER
00928 353F1 F00AD  4356 +          OR DB,B,RAME,R1 & NOP RBAU+1    ;OR IN 12 LSBS
00929 0091F F00AC  4357            ADD AQ,FOUTU,R1,RAME & NOP RBAU ;INCR. IT + RECORD LENGTH AND...
0092A 0011F F00AD  4358            ADD AQ,FOUTL,R1,RAME & NOP RBAL ;SAVE IT
                   4359
                   4360    SMBOK12A: JSR.P SMBOOK20                ;COMPUTE NEXT RECORD JUMP
0092B E0403 1F2AC  4360 +          SETPGE SMBOOK20_SHR_H#B & JSRP SMBOOK20
0092C 4505E F4000  4361            TSTRBIT R14,4000                ;READ/WRITE STARTED?
0092D 420FF 13968  4362            RESETQ & CJSR NALUZ,DMACHKB     ;YES, CHECK SM DMA ADDRESS
                   4363            BNZ.P READ62                    ;IF CHECK FAILURE, EQUIP. CHECK
0092E E0400 33CAD  4363 +          SETPGE READ62_SHR_H#B & BNZP READ62
0092F 3535E F4000  4364            SETRBIT R14,4000                ;SET READ/WRITE STARTED FLAG
                   4365    SMBOK12B: JSR.P RDSMADDR                ;Compute next SM-addr.
00930 E0400 1F4C8  4365 +          SETPGE RDSMADDR_SHR_H#B & JSRP RDSMADDR
00931 C2002 FFFFF  4366            RD HSSMA,RDINTR,R2
00932 45052 F0010  4367            TSTRBIT R2,10                   ;Did the HSSM go "busy"?
00933 31110 33136  4368            OR AB,FOUTL,R1,R0 & BOZ $+3     ;If not, continue
                   4369            JSR.P CHKBUSY                   ;Wait for "busy" to go away and...
00934 E0400 1F535  4369 +          SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
00935 FFFFF E0930  4370            JMP SMBOK12B                    ;Reload the next SM-addr.
                   4371    ;
                   4372            CJSR.P ALUZ,SMMAX               ;IF 0, SET CHECK ADDR. TO MAX.
00936 E0403 1372A  4372 +          SETPGE SMMAX_SHR_H#B & CJSRP ALUZ,SMMAX
00937 3491F F0351  4373            ALDRAMU R1,NEXTSMU              ;SAVE THE MSB'S AND...
00938 3410F AF352  4374            ALDREG R0,RAME & RTS NEXTSML    ;LSB'S FOR CHECKING; RETURN
                   4402
                   4403    ;   CHANNEL (U-SIDE) BOOKKEEPING
                   4404    ;
00939 5535F F4000  4405    UBOOK:  RSTRBIT R15,4000                ;RESET U-INITIATED FLAG
0093A 45055 F0800  4406            TSTRBIT R5,800                  ;WRITE OPERATION?
0093B FFFFF 33959  4407            BNZ UBOOK03                     ;IF IT IS, JUMP
```

Addr Line CDC O.S. - BOOKKEEPING

```
                4408    ;
                4409    ;   U-SIDE READ BOOKKEEPING
                4410    ;
0093C 1938A FFFFF  4411    UBOOK01:  SUBA AB,B,R8,R10                ;U_COUNT - U_XFER._COUNT = U_COUNT
0093D 34382 33940  4412              RTOR     R8,R2 & BNZ $+3         ;IF U_COUNT NOT 0, JUMP
0093E 3535E F0400  4413              SETRBIT R14,400                  ;SET U-COMPLETE FLAG
0093F FFFFF E0946  4414              JMP UBOOK01A
                4415    ;
00940 0F3F0 F0206  4416              ADD DZC,B,RAME,R0 & NOP URELREC   ;GET U-SIDE RELATIVE REC.+ 1
00941 2D0F0 F02D0  4417              SUBB DB,Q,RAME,R0 & NOP MAXREL    ;IS MAX. RELATIVE > CURRENT?
00942 FFFFF 34145  4418              CJMP AGB,$+3                     ;YES, JUMP
00943 3710F F0206  4419              RESETRAM URELREC                 ;ELSE, RESET U-SIDE RELATIVE REC...
00944 433F6 E1F24  4420              RESETR R6 & JMP SETUMAX          ;AND THE WORD NUMBER
00945 3410F F0206  4421              ALDRAM R0,URELREC                ;SAVE NEXT U-SIDE RELATIVE REC.
                4422    ;
00946 01386 FFFFF  4423    UBOOK01A: ADD AB,B,R8,R6                   ;UXFER + WORD NO.= WORD NO.
                4424              JSR.P NXTUADDR                   ;COMPUTE NEXT U-START ADDRESS...
00947 E0400 1F4B0  4424 +          SETPGE NXTUADDR_SHR_H#B & JSRP NXTUADDR
                4425              JSR.P NXTUDMA                    ;AND SAVE IT FOR CHECKING
00948 E0403 1F55F  4425 +          SETPGE NXTUDMA_SHR_H#B & JSRP NXTUDMA
00949 2D1FA F02CD  4426    UBOOK02:  SUBB DB,FOUTL,RAME,R10 & NOP RECLNTH ;REC. LENGTH = OR > U COUNT?
0094A 343A8 35961  4427              RTOR R10,R8 & CJMP NALB,UBOOK05  ;IF YES, JUMP; U XFER.CNT.= UCNT
0094B 373F8 F02CD  4432              RDRAMB R8,RECLNTH                ;ELSE, U_XFER. COUNT = REC. LENGTH
0094C 4505E F0008  4433              TSTRBIT R14,8                    ;HOME ADDR. FLAG?
0094D FFFFF 3314F  4434              BOZ $+2                          ;IF NOT, JUMP
0094E 05348 F0002  4435              ADD DB,B,MSK1,R8 & NOP D#2       ;INCR. RECORD LENGTH +2
                4436    ;
0094F 273F0 F0206  4437              SAB DZ,B,RAME,R0 & NOP URELREC   ;U-SIDE RELATIVE RECORD -1
00950 FFFFF 31152  4438              CJMP CARRY,$+2                   ;IF A POSITIVE NO., JUMP
00951 273F0 F02D0  4439              SAB DZ,B,RAME,R0 & NOP MAXREL    ;GET MAX. RELATIVE RECORD -1
00952 651F0 F0029  4440              XOR DB,FOUTL,RAME,R0 & NOP RELREC ;U-RELATIVE = NEXT SM-RELATIVE?
00953 FFFFF 33961  4441              BNZ UBOOK05$                     ;IF NOT, CONTINUE
                4442    ;
00954 4505F F1000  4443              TSTRBIT R15,1000                 ;ADV. CYLINDER FLAG?
00955 FFFFF 33961  4444              BNZ UBOOK05$                     ;YES, CONTINUE
                4445    ;
00956 37353 F003A  4446              RDMSKB R3,WAITOCON               ;MPC WAIT-TO-CONTINUE JUMP
                4447              LDAR.P SMBOOK25
00957 E0403 5F2ED  4447 +          SETPGE SMBOOK25_SHR_H#B & LDARP SMBOOK25
00958 FFFFF E1AE2  4448              JMP SMBOK23A                     ;PROCESS 'STOP READ'
                4449    ;
                4450    ;   U-SIDE WRITE BOOKKEEPING
                4451    ;
                4452    UBOOK03:  EQU $
00959 377F0 F02CF  4459              OR DZ,LSBZ,RAME,R0 & NOP BUFCAP  ;R0 = 1/2 BUFFER CAPACITY
0095A 337F0 FFFFF  4460              OR ZB,LSBZ,,R0                   ; X2 = BUFFER CAPACITY
0095B 1938A FFFFF  4461              SUBA AB,B,R8,R10                 ;U-COUNT - UXFER = U-COUNT
0095C 01068 33160  4462              ADD AB,Q,R6,R8 & BOZ UBOOK04     ;IF 0, JUMP; ST. WORD + UXFER = Q
```

```
 Addr            Line CDC O.S. - BOOKKEEPING

00950 28100 FFFFF  4463            SUBB AQ,FOUTL,R0,ZERO            ;IF Q IS < BUFFER CAPACITY...
00950 323F6 34161  4464            QTOR R6 & CJMP AGB,UBOOK05       ;SET START WORD = Q & JUMP
00950 433F6 E0961  4465            RESETR R6 & JMP UBOOK05          ;ELSE, START WORD = 0
                   4466    ;
00960 3535E F0400  4467    UBOOK04: SETRBIT R14,400                  ;SET U-COMPLETE
                   4468    UBOOK05: JSR.P CHKBUSY                    ;CHECK FOR HSSM BUSY
00961 E0400 1F535  4468 +          SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
00962 373F0 F02E7  4469            RDRAMB R0,SAVXMCJ                ;GET SAVED JUMP COMMAND AND...
00963 91F0F FFFFF  4470            WRTL HSSM,LDXMCJ,R0              ;RELOAD IT
00964 C2000 FFFFF  4471            RD HSSMA,RDINTR,R0
00965 40100 FFFFF  4472            AND AQ,FOUTL,R0,ZERO             ;CHECK AGAIN FOR BUSY
00966 FFFFF A37FF  4473            CRTS ALUZ                        ;IF NOT, RETURN
00967 FFFFF E0961  4474            JMP UBOOK05                      ;ELSE, WAIT FOR NOT BUSY
                   4475
                   4476
00968 FFFFF E1D20  4477    DMACHKB: JMP DMACHK
                   4478
                   4479
                   4481    TITLE CDC O.S. - DRIVE SELECTION
```

Addr Line CDC O.S. - DRIVE SELECTION

```
4490
4491    ;     THE REGISTERS FOR DRIVE SELECTION ARE:
4492    ;
4493    ;       R0,R1,R2,Q & IREG = SCRATCH
4494    ;       R3 = BIT FOR THIS DRIVE (24-31), RECORD LENGTH FLAGS
4495    ;       R4 = BIT FOR THIS DRIVE (0-23), MAX. DRIVE CYLINDER
4496    ;       R5 = DRIVE STATUS, MAX. HEAD
4497    ;       R6 = SCRATCH
4498    ;       R7 = TRANSFER LENGTH (FROM FUNCT. WORD)
4499    ;       R8 = Scratch
4500    ;       R9 & R11 = NOT USED (Reserved)
4501    ;       R10 = DRIVE TYPE POINTER
4502    ;       R12 = CURRENT COMMAND CODE
4503    ;       R13 = DRIVE NO.
4504    ;       R14,R15 = FLAG REGISTERS
4505
4506    ;
4507    ;
4508
4511    JDRIVE:   EQU H#100                     ;TYPE 'J' SWALLOW-5
4516    DFECT:    EQU 0                         ;NO DEFECT TRACK INFO.
4518
4519
4520    ;     MAXIMUM DRIVE CYLINDER
4521    ;
4522    MAXCYL0:  EQU D#2047                    ;SPARE
4523    MAXCYL1:  EQU D#1633                    ;CDC SABRE-5 DRIVE
4524    MAXCYL2:  EQU D#843                     ;CDS 513-MB. DRIVE
4525    MAXCYL3:  EQU D#1645                    ;FUJI M2333K DRIVE
4526    MAXCYL4:  EQU D#743                     ;FUJI SWALLOW-5 DRIVE
4527    MAXCYL5:  EQU D#623                     ;FUJI SWALLOW-4 DRIVE
4528    MAXCYL6:  EQU D#2433                    ;CDC 368-MB.
4529    MAXCYL7:  EQU D#2431                    ;CDC 368-MB. WITH DEFECTS
4530
4531    ;    HSSM DRIVE TYPE POINTER TABLE
4532
4533    POINT0:   EQU H#00                      ;
4534    POINT1:   EQU H#36+DFECT                ;CDC SABRE-5 DRIVE (TEMP. POINTER)
4535    POINT2:   EQU H#24+DFECT                ;CDS 513-MB. DRIVE
4536    POINT3:   EQU H#C8+TWOFER               ;FUJITSU 337-MB. DRIVE
4538    POINT4:   EQU H#DA                      ;FUJITSU SWALLOW-5 (HARD SECTORED)
4542    POINT5:   EQU H#12                      ;FUJITSU SWALLOW-4 DRIVE
4543    POINT6:   EQU H#36+TWOFER               ;CDC-368 DRIVE (MPI PABA2)
4544    POINT7:   EQU H#36+TWOFER+DFECT         ;CDC-368 DRIVE WITH DEFECT INFO.
4545
```

Addr Line CDC O.S. - DRIVE SELECTION

```
                    4546
                    4560
00969 343D1 FFFFF   4561    SEL00:    RTOR R13,R1                     ;LOGICAL DRIVE NO. TO R1
                    4562              JSR.P DRIVSTAT                  ; GET DRIVE STATE
0096A E0404 1F000   4562 +            SETPGE DRIVSTAT_SHR_H#B & JSRP DRIVSTAT
00968 3418F F0263   4563              ALDRAM R8,PRSSTATE              ; SAVE DRIVE STATE
                    4564              JSR.P LOGICAL                   ;LOGICAL DRIVE NO. TO Q
0096C E0400 1F1F9   4564 +            SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
                    4565    ;
0096D 323F0 1F329   4566              QTOR R0 & JSR SEL40             ;COMPUTE THE BIT FOR THIS DRIVE
0096E 37352 F006F   4567              RDMSKB R2,BITILD                ;LOC. OF ILD BIT FOR THIS DRIVE
0096F 373F1 F00BA   4568              RDRAMB R1,ILDFLG                ;GET ILD FLAGS
00970 45351 F0004   4569              MSKRBIT R1,4                    ;ILD RUNNING?
00971 FFFFF 3317A   4570              BOZ SEL00A                      ;IF NOT RUNNING, JUMP
                    4571    ;
00972 3714E F0054   4572              MOVE MSK1,IREG & NOP SKPEND     ;GET SEEK COMPLETE PENDING...
                    4573              JSR.P SMBRD                     ;BITS FOR ALL DRIVES
00973 E0400 1F03B   4573 +            SETPGE SMBRD_SHR_H#B & JSRP SMBRD
00974 41104 FFFFF   4574              AND AB,FOUTL,R0,R4              ;IF THERE IS AN ON-LINE SEEK...
00975 40131 33ADF   4575              AND AQ,FOUTL,R3,R1 & BNZ SEL20A ;COMPLETE PENDING ON THE ILD...
00976 3403F 33ADF   4576              RTOQ R3 & BNZ SEL20A            ;DRIVE, SET INTERVENTION REQD...
00977 34340 FFFFF   4577              RTOR R4,R0                      ;TO THE ILD
00978 3412E 1F0CE   4578              ALDREG R2,IREG & JSR SMBWRTA    ;SET THE BIT FOR THE ILD DRIVE
00979 433F2 E0985   4579              RESETR R2 & JMP SEL00B
                    4580    ;
                    4581    SEL00A:   TSTMEQU  50,SEN00               ;Sense 0 = Intervention & Equip. chk?
0097A 373F0 F0000   4581 +            RDRAMB   R0,SEN00
0097B 65150 F0050   4581 +            TSTREQU  R0,50
0097C FFFFF 33308   4583              BOZ      SEL28                  ;$Yes, Equipment Check
0097D 3412E 1F0CD   4587              ALDREG R2,IREG & JSR SMBRDA     ;GET THE BIT FOR THE ILD DRIVE
0097E 41140 33181   4588              AND AB,FOUTL,R4,R0 & BOZ $+3    ;IF AN ILD IS RUNNING ON...
0097F 40130 33AED   4589              AND AQ,FOUTL,R3,R0 & BNZ SEL21  ;THE DESIRED DRIVE,
00980 433F2 33AED   4590              RESETR R2 & BNZ SEL21           ;SEND INTERVENTION REQUIRED
                    4591    ;
00981 4505E F0010   4592              TSTRBIT R14,10                  ;READ COUNT FLAG?
00982 FFFFF 1352E   4593              CJSR ALUZ,CLRSENS               ;If not, clear Sense Bytes
                    4594    ;
00983 FFFFF 1F1D5   4600              JSR SEL07                       ;RESET SEEK COMP. PENDING
                    4601              JSR.P CLRSECT1                  ;CLEAR SAVED SECTOR THIS DRIVE
00984 E0403 1F368   4601 +            SETPGE CLRSECT1_SHR_H#B & JSRP CLRSECT1
```

Addr Line CDC O.S. -- DRIVE SELECTION

```
                4603    SEL00B:  JSR.P LOGICAL                    ;GET LOGICAL DRIVE NO.
00985 E0400 1F1F9 4603 +         SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
00986 323FD FFFFF 4604           QTOR R13                         ;MOVE IT TO R13
                4605    ;
00987 E0082 FFFFF 4606    SEL01: CTL SMC,HSSMRST                  ;RESET HSSM INTERRUPTS
00988 E100B FFFFF 4607           CTL HSSM,RSTUSEL                 ;DE-SELECT PRESENT DRIVE
00989 4535D F003F 4611           MSKRBIT R13,3F                   ;ISOLATE LOGICAL DRIVE IN R13
0098A FFFFF 1F333 4612           JSR STRING                       ;DRIVE STRING NO. FOR SENSE
                4613             JSR.P DOUBLE                     ;DOUBLE DR. NO. IF NEEDED
0098B E0403 1F125 4613 +         SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
0098C 4505E F0800 4614           TSTRBIT R14,800                  ;SECOND DRIVE FLAG?
0098D 340DF 3318F 4615           RTOQ R13 & BOZ $+2               ;IF NOT, JUMP; PHYS. DR. TO Q
0098E 3535D F0001 4616           SETRBIT R13,1                    ;INCR. DRIVE NO.+1
                4617    ;
0098F B2200 F0000 4618           WRTE HSSMA,LDDRBO & NOP 0        ;CLEAR DATA BUS OUT REG.
00990 341DF F000D 4619           ALDRAM R13,SEN113                ;PHYSICAL DRIVE FOR SENSE
                4620             SETMBIT H#E3,SEN115              ;SELECT TAG to Sense 15, Format 1
00991 37350 F00E3 4620 +         RDMSKB   R0,H#E3
00992 3410F F000F 4620 +         ALDRAM   R0,SEN115
00993 46346 F0070 4621           AND DQ,B,MSK1,R6 & NOP H#70      ;CARD CAGE BITS (CHALLNGR) TO R6
                4622    ;
                4623    ; SELECT THE DRIVE
                4624    ;
00994 91EDF 4F7FF 4625           WRTL HSSM,LDDRUSEL,R13 & PSHLD H#7FF ;SET UNIT SEL. REG.
00995 C1045 FFFFF 4626           RD HSSM,RDDRST1,R5
00996 C1045 B819D 4627           RD HSSM,RDDRST1,R5 & CJPOP EVCNTZ,SEL02 ;READ DRIVE STATUS
00997 E1007 83FFF 4628           CTL HSSM,SETUSEL & CLPCNT NALUZ  ;IF NONE, JUMP BACK; SELECT DRIVE
                4629    ;
00998 91E6F FFFFF 4630           WRTL HSSM,LDDRUSEL,R6            ;LOAD CARD CAGE SELECT ONLY
00999 37358 F0098 4631           RDMSKB R8,98                     ;MSB's of 2-second timer...
0099A D0FF8 C0004 4632           RDXFR SMC,,R8 & LDCT 4
0099B 35358 F0968 4633           SETRBIT R8,968                   ;Plus LSB's into R8
0099C C1045 9819C 4634           RD HSSM,RDDRST1,R5 & CJPCNT EVCNTZ,$ ;READ DRIVE STATUS
                4635    ;
0099D 45055 F0020 4636    SEL02: TSTRBIT R5,SELECT                ;CHECK FOR UNIT SELECTED
0099E FFFFF 332DE 4637           BOZ SEL20                        ;IF NOT SELECTED, INTERVENTION
0099F FFFFF 1F337 4638           JSR PHYSAD                       ;COMPUTE PHYS. ADDR.
009A0 4505F F2000 4639           TSTRBIT R15,2000                 ;POLLING FLAG SET?
009A1 FFFFF 131E0 4640           CJSR ALUZ,SEL09                  ;IF NOT, COMMAND CHECK
009A2 45055 F0010 4641           TSTRBIT R5,WRITPROT              ;DRIVE "WRITE PROTECT"?
009A3 FFFFF 331A5 4642           BOZ SEL02A                       ;IF NOT, JUMP
009A4 3535F F0100 4643           SETRBIT R15,100                  ;SET READ ONLY FLAG
                4644    ;
009A5 3714F F0008 4645    SEL02A: MOVE MSK1,RAME & NOP H#8        ;SENSE 8, FORMAT 1 = ON LINE
009A6 3415F F000E 4646           ALDRAM R5,SEN114                 ;SAVE DRIVE STATUS FOR SENSE
                4647    ;
009A7 45055 F0040 4661           TSTRBIT R5,BUSY                  ;Drive Busy?
009A8 133F8 331AB 4662           DECRB R8 & BOZ SEL02C            ;No, continue; Decr. Busy timer
```

Addr Line CDC O.S. - DRIVE SELECTION

```
009A9 FFFFF 339CB  4663            BNZ SEL05                          ;If timer not expired, jump
009AA FFFFF E0B03  4664            JMP SEL25                          ;Otherwise, Equip. Check
                   4665    ;
009AB 37341 4F001  4666    SEL02C: RDREGB MSK1,R1 & PSHLD READY       ;SET 'READY' MASK
009AC C1045 887FF  4667            RD HSSM,RDDRST1,R5 & CLPCNT EVCNTZ ;READ DRIVE STATUS
009AD 45055 F0008  4668            TSTRBIT R5,FAULT                   ;DRIVE FAULT?
009AE 41015 13B0A  4669            AND AB,Q,R1,R5 & CJSR NALUZ,SEL30  ;YES, ATTEMPT TO RESET IT
009AF FFFFF 332DE  4670            BOZ SEL20                          ;IF DRIVE NOT READY, INTERVENTION
                   4671    ;
009B0 3415F F000E  4672    SEL03:  ALDRAM R5,SEN114                   ;SAVE DRIVE STATUS FOR SENSE
                   4673            SETMBIT 21,SEN110                  ;SET SENSE: ACCESS READY/HEAD LOAD
009B1 37350 F0021  4673 +          RDMSKB  R0,21
009B2 3410F F000A  4673 +          ALDRAM  R0,SEN110
009B3 3710F F0010  4674            RESETRAM SEN121                    ;CLEAR SENSE BYTE 21, FORMAT 1
009B4 3710F F000B  4675            RESETRAM SEN111                    ;CLEAR SENSE BYTE 11, FORMAT 1
                   4676    ;
009B5 4505F F2000  4677            TSTRBIT R15,2000                   ;CHECK FOR POLLING
009B6 603FF A3FFF  4678            XOR AQ,B,R15,R15 & CRTS NALUZ      ;IF POLLING, RETURN
                   4679    ;
009B7 771FE F002D  4680            XNOR DZ,FOUTL,RAME,IREG & NOP EFALLOW ;GET THE BIT FOR THIS HIO
009B8 3705F F09B7  4681            RDMSKQ $-1
009B9 0613E 5F1BF  4682            ADD DQ,FOUTL,VPE,IREG & LDAR SEL04 ;COMPUTE JUMP LOCATION
009BA FFFFF 6F7FF  4683            JIR
                   4684    ;
009BB 37342 5F884  4685            RDREGB MSK1,R2 & JAR ATTN4         ;LOC. OF ATTN. PENDING HAIO-4
009BC 37342 5F87E  4686            RDREGB MSK1,R2 & JAR ATTN3         ;LOC. OF ATTN. PENDING HAIO-3
009BD 37342 5F878  4687            RDREGB MSK1,R2 & JAR ATTN2         ;LOC. OF ATTN. PENDING HAIO-2
009BE 37342 F0072  4688            RDREGB MSK1,R2 & NOP ATTN1         ;LOC. OF ATTN. PENDING HAIO-1
                   4689    ;
009BF 373F6 F00BA  4690    SEL04:  RDRAMB R6,ILDFLG                   ;GET ILD FLAGS
009C0 3412E 1F0CD  4691            ALDREG R2,IREG & JSR SMBRDA        ;GET ATTN. PENDING BITS THIS HAIO
009C1 41140 331E4  4692            AND AB,FOUTL,R4,R0 & BOZ SEL10     ;IF ATTENTION PENDING...
009C2 40130 339C4  4693            AND AQ,FOUTL,R3,R0 & BNZ $+2       ;ON THIS DRIVE (0-23), JUMP
009C3 FFFFF 331E4  4694            BOZ SEL10                          ;IF NOT DRIVE 24-31, JUMP
                   4695    ;
009C4 45156 F0004  4696            TSTBIT R6,4                        ;CHECK ILD RUNNING
009C5 51340 339E4  4697            MASK AB,B,R4,R0 & BNZ SEL10        ;IF RUNNING, JUMP
009C6 5003F FFFFF  4698            MASK AQ,Q,R3                       ;DELETE ATTENTION PENDING...
009C7 3412E 1F0CE  4699            ALDREG R2,IREG & JSR SMBWRTA       ;FOR THIS DRIVE ON THIS HAIO
009C8 3535D F1000  4700            SETRBIT R13,1000                   ;CHAN. STATUS = BUSY
009C9 37352 F0019  4701            RDMSKB R2,19                       ;FILE STATUS NOT AS EXPECTED
009CA 433F0 E0B78  4702            RESETR R0 & JMP POLL05C            ;SEND ATTN./BUSY STATUS
                   4703    ;
009CB 4505E F0020  4704    SEL05:  TSTRBIT R14,20                     ;SPECIAL FLAG SET?
009CC FFFFF 339D0  4705            BNZ SEL05A                         ;IF YES, WAIT FOR NOT BUSY
                   4706            TSTMBIT 4,ILDFLG                   ;ILD Running?
009CD 373F0 F00BA  4706 +          RDRAMB  R0,ILDFLG
009CE 45150 F0004  4706 +          TSTBIT  R0,4
009CF FFFFF 331D2  4707            BOZ SEL05B                         ;If not, jump
009D0 C1045 FFFFF  4708    SEL05A: RD HSSM,RDDRST1,R5                 ;Read Drive status...
009D1 C1045 E09A6  4710            RD HSSM,RDDRST1,R5 & JMP SEL02A+1  ;$twice and check again for Busy
```

Addr Line CDC O.S. - DRIVE SELECTION

```
                    4714    ;
009D2 4505F F2000   4715    SEL05B:   TSTRBIT R15,2000                ;TEST FOR POLLING FLAG
009D3 603FF A3FFF   4716              XOR AQ,B,R15,R15 & CRTS NALUZ     ;IF SET, RESET IT & RETURN
009D4 FFFFF E09D0   4718              JMP     SEL05A                    ;$Otherwise, wait for not Busy
                    4731
                    4732    ;    THIS SUBROUTINE WILL CLEAR THE SEEK COMPLETE PENDING BIT FOR FOR
                    4733    ;THIS DRIVE ON ALL HIO'S.
                    4734
009D5 37352 F0054   4735    SEL07:    RDMSKB R2,SKPEND                ;LOC. OF SEEK COMP. PENDING
009D6 FFFFF 4F004   4736              PSHLD 4                         ;5 LOCATIONS TO CHECK
009D7 FFFFF 1F1DC   4737              JSR DELABIT                     ;CLEAR SEEK PEND. THIS DRIVE
009D8 05342 88003   4738              ADD DB,B,MSK1,R2 & CLPCNT EVCNTZ,D#3 ;IF NOT DONE, CHECK NEXT
                    4739    ;
009D9 37352 F03FC   4740              RDMSKB R2,RPSPEND               ;CLEAR R.P.S. PENDING BIT...
009DA FFFFF 1F1DC   4741              JSR DELABIT                     ;FOR THIS DRIVE
                    4742    ;
009DB 37352 F02B5   4743    SEL07A:   RDMSKB R2,DRONCYL               ;LOC. OF DRIVE ON CYLINDER
                    4744    ;
009DC 3412E 1F0CD   4745    DELABIT:  ALDREG R2,IREG & JSR SMBRDA     ;GET THE BITS
009DD 51340 A37FF   4746              MASK AB,B,R4,R0 & CRTS ALUZ     ;IF NONE, RETURN
009DE 5003F FFFFF   4747              MASK AQ,Q,R3                    ;ELSE, CLEAR THE BIT FOR...
009DF 3412E E0044   4748              ALDREG R2,IREG & JMP SMBWRT     ;THIS DRIVE & RETURN
                    4749
009E0 6505C F0015   4750    SEL09:    TSTRMSK R12,15                  ;CHECK "WRITE H.A.& R0" COMMAND
009E1 FFFFF A3FFF   4751              CRTS NALUZ                      ;IF NOT, RETURN
009E2 37352 F0001   4752              RDMSKB R2,1                     ;SENSE 7: INVALID COMMAND
                    4753              JPOP.P COMREJ1A                 ;SEND COMMAND REJECT
009E3 E0400 BF102   4753 +            SETPGE COMREJ1A_SHR_H#B & JPOPP COMREJ1A
```

Addr Line CDC O.S. - DRIVE SELECTION

```
                    4755
                    4756    ;    THIS PORTION OF THE SELECTION ROUTINE WILL SET UP ALL THE CONSTANTS AND
                    4757    ;LIMITS NEEDED IN OTHER PARTS OF THE PROGRAM.  THEY WILL BE SET UP FROM THE
                    4758    ;FLAG BITS STORED IN THE SMB MEMORY FOR EACH DRIVE.  IF NO FLAGS ARE STORED
                    4759    ;FOR THIS DRIVE, THE PROGRAM WILL ATTEMPT TO READ THE FLAGS FROM THE HEADER
                    4760    ;FIELD RECORDED WITH EACH RECORD ON THE DRIVE.  IF SUCCESSFUL, THE FLAGS
                    4761    ;WILL BE SAVED FOR FUTURE USE AND THE CONSTANTS AND LIMITS WILL BE COMPUTED.
                    4762    ;IF UNSUCCESSFUL, UNIT CHECK STATUS WITH SENSE BYTE 0 = EQUIP. CHECK,
                    4763    ;SENSE BYTE 1 = INVALID TRACK FORMAT AND SENSE 7 = 1E (UNABLE TO DETERMINE
                    4764    ;DRIVE TYPE AT INITIAL SELECTION) WILL BE GENERATED. RECOV. ACTION WILL BE 'A'.
                    4765    ;
                    4766    SEL10:    EQU $
009E4 FFFFF 1F342   4778              JSR FLAGLOC                    ;LOC. OF DRIVE FLAGS
009E5 3710F F0025   4779    SEL10A:   RESETRAM SEN216                ;CLEAR SENSE 16, FORMAT 2
                    4780              SVWD     R7,SAVXFRU            ;Temp. save R7
009E6 3497F F024A   4780 +            ALDRAMU  R7,SAVXFRU             ;SAVE 12 MSBS
009E7 3417F F024B   4780 +            ALDRAM   R7,SAVXFRU+1           ;SAVE 12 LSBS
009E8 373D8 C0001   4781              RDREGB RAMI,R8 & LDCT 1        ;GET THE FLAGS INTO R8
009E9 34586 3329E   4782              OR AZ,RSBZ,R8,R6 & BOZ SEL15   ;IF NONE, ATTEMPT TO READ HEADER
009EA 370FF F03E5   4783              RDRAMQ LASTFLG                 ;GET LAST FLAG
009EB 60180 3329E   4784              XOR AQ,FOUTL,R8,ZERO & BOZ SEL15 ;IF NONE, READ HEADER
009EC FFFFF 3329B   4785              BOZ      SELRTN                ;If this flag = last flag, return
009ED 3418F F03E5   4786    SEL10B:   ALDRAM R8,LASTFLG              ;ELSE, THIS FLAG IS LAST FLAG
                    4787    ;
009EE 335F6 981EE   4789              OR ZB,RSBZ,,R6 & CJPCNT EVCNTZ,$ ;<RECORD LENGTH X2 INTO POSITION
009EF 34363 FFFFF   4790              RTOR R6,R3                     ;<AND INTO R3
009F0 45353 F0006   4791              MSKRBIT R3,6                   ;<ISOLATE RECORD LENGTH X2 ONLY
                    4846              SETMBIT  D#116,RECLNTH         ;$Record Length = 116 words
009F1 37350 F0074   4846 +            RDMSKB   R0,D#116
009F2 3410F F02CD   4846 +            ALDRAM   R0,RECLNTH
009F3 433F2 FFFFF   4847              RESETR   R2
009F4 433FA FFFFF   4848              RESETR R10
009F5 37351 F007D   4849              RDMSKB R1,D#125                ;$ 125 RELATIVE RECORDS
009F6 37350 F0E29   4850              RDMSKB R0,D#3625               ;$ 1/4 BUFFER CAPACITY
009F7 3410F F02CF   4851              ALDRAM R0,BUFCAP               ;$SAVE IT
009F8 3411F F02D0   4895              ALDRAM R1,MAXREL               ;SAVE MAX. RELATIVE RECORD
                    4896    ;
009F9 45058 F000E   4897              TSTRBIT R8,H#E                 ;ISOLATE DRIVE TYPE X2 INTO Q
009FA 325F1 FFFFF   4898              OR       ZQ,RSBZ,,R1           ;Drive type to R1
```

```
 Addr            Line CDC O.S. - DRIVE SELECTION

009FB 436F4 FFFFF  4900            AND ZB,LSBQZ,,R4                ;< Q X 2 = X4 INTO Q
009FC 327F4 FFFFF  4901            OR ZQ,LSBZ,,R4                  ;< Q X 2 = X8 INTO R4
009FD 0004F 5F269  4902            ADD AQ,Q,R4 & LDAR SEL14        ;< Q + R4 = X12
009FE 0001F FFFFF  4911            ADD     AQ,Q,R1                 ;Add Drive type
009FF 0615E F0A01  4912            ADD DQ,FOUTL,MSK2,IREG & NOP $+D#2 ;DO A JUMP ACCORDING TO DRIVE...
00A00 335F3 6F7FF  4914            OR ZB,RSBZ,,R3 & JIR            ;<type X13; rt. shift lngth. flags
                   4918   ;
                   4919   ;(0) SPARE DRIVE TYPE
00A01 37354 F07FF  4920            RDMSKB R4,D#2047                ;SPARE   MAX. USABLE CYL.=
00A02 37350 F07FF  4921            RDMSKB R0,MAXCYL0               ;SPARE   MAX. DRIVE CYL.=
00A03 37355 F003F  4922            RDMSKB R5,D#63                  ;SPARE   MAX. HEAD =
00A04 37357 F0000  4923            RDMSKB   R7,D#0                 ;Spare   Max. Sector =
00A05 FFFFF FFFFF  4925            NOP
00A06 05053 F0A0A  4926            ADD DB,Q,MSK2,R3 & NOP $+D#4    ;<JUMP LOC. ACCORDING...
00A07 321FE FFFFF  4927            QLDREG IREG                     ;<TO RECORD LENGTH
00A08 37353 F0FFF  4929            RDMSKB R3,H#FFF                 ;$ 1/4 WORDS PER TRACK AT 116
00A09 0534A 6F000  4933            ADD DB,B,MSK1,R10 & JIR POINT0  ;<SPARE   DRIVE TYPE POINTER
00A0A 37341 5F800  4939            RDREGB MSK1,R1 & JAR D#0        ;INVALID RECORD LENGTH
00A0B 37341 5F87F  4941            RDREGB MSK1,R1 & JAR D#127      ;$SPARE   MAX. RECORD (116 WORDS)
00A0C 37341 5F800  4942            RDREGB MSK1,R1 & JAR 0          ;$INVALID RECORD LENGTH
00A0D 37341 5F800  4943            RDREGB MSK1,R1 & JAR 0          ;$INVALID RECORD LENGTH
                   4961   ;
                   4962   ;(1) CDC SABRE 5 DRIVE
00A0E 37354 F0634  4963            RDMSKB R4,D#1588                ;SABRE 5 USABLE CYL.=1588
00A0F 37350 F0661  4964            RDMSKB R0,MAXCYL1               ;SABRE 5 MAX. DRIVE CYL.=1633
00A10 37355 F000E  4965            RDMSKB R5,D#14                  ;SABRE 5 HEAD = 14
00A11 37357 F0000  4966            RDMSKB   R7,D#0                 ;Sabre 5 R.P.S. not supported
00A12 FFFFF FFFFF  4968            NOP
00A13 05053 F0A17  4969            ADD DB,Q,MSK2,R3 & NOP $+D#4    ;<JUMP LOC. ACCORDING...
00A14 321FE FFFFF  4970            QLDREG IREG                     ;<TO RECORD LENGTH
00A15 37353 F0797  4972            RDMSKB R3,797                   ;$ 1/4 WORDS PER TRACK AT 116
00A16 0534A 6F036  4976            ADD DB,B,MSK1,R10 & JIR POINT1  ;<SABRE 5 DRIVE TYPE POINTER
```

Addr Line CDC O.S. - DRIVE SELECTION

```
00A17 37341 5F800  4982            RDREGB MSK1,R1 & JAR D#0          ;INVALID RECORD LENGTH
00A18 37341 5F843  4984            RDREGB MSK1,R1 & JAR D#67         ;$SABRE 5 MAX. RECORD (116 WORDS)
00A19 37341 5F800  4985            RDREGB MSK1,R1 & JAR 0            ;$INVALID RECORD LENGTH
00A1A 37341 5F800  4986            RDREGB MSK1,R1 & JAR 0            ;$INVALID RECORD LENGTH
                   5004    ;
                   5005    ;(2) CDS 513-MB. DRIVE
00A1B 37354 F0347  5006            RDMSKB R4,D#839                   ;CDS 513 MAX. USABLE CYL.=839
00A1C 37350 F034B  5007            RDMSKB R0,MAXCYL2                 ;CDS 513 MAX. DRIVE CYL.= 843
00A1D 37355 F0012  5008            RDMSKB R5,D#18                    ;CDS 513 MAX. HEAD = 18
00A1E 433F7 FFFFF  5009            RESETR  R7                        ;CDS 513 R.P.S. not supported
00A1F FFFFF FFFFF  5011            NOP
00A20 05053 F0A24  5012            ADD DB,Q,MSK2,R3 & NOP $+D#4      ;<JUMP LOC. ACCORDING...
00A21 321FE FFFFF  5013            QLDREG IREG                       ;<TO RECORD LENGTH
00A22 37353 F0658  5015            RDMSKB R3,658                     ;$ 1/4 WORDS PER TRACK AT 116
00A23 0534A 6F024  5019            ADD DB,B,MSK1,R10 & JIR POINT2    ;<CDS-513 DRIVE TYPE POINTER
00A24 37341 5F800  5025            RDREGB MSK1,R1 & JAR D#0          ;INVALID RECORD LENGTH
00A25 37341 5F836  5027            RDREGB MSK1,R1 & JAR D#54         ;$CDS-513 MAX. RECORD (116 WORDS)
00A26 37341 5F800  5028            RDREGB MSK1,R1 & JAR 0            ;$INVALID RECORD LENGTH
00A27 37341 5F800  5029            RDREGB MSK1,R1 & JAR 0            ;$INVALID RECORD LENGTH
                   5047    ;
                   5048    ;(3) FUJITSU 337-MB. DRIVE (M2333K)
00A28 37354 F0669  5049            RDMSKB R4,D#1641                  ;FUJI. 337 MAX. USABLE CYL.=1641
00A29 37350 F066D  5050            RDMSKB R0,MAXCYL3                 ;FUJI. 337 MAX. DRIVE CYL.=1645
00A2A 37355 F0009  5051            RDMSKB R5,D#9                     ;FUJI. 337 MAX. HEAD = 9
00A2B 433F7 FFFFF  5052            RESETR  R7                        ;Fuji. 337 R.P.S. not supported
00A2C 37372 FFFFF  5054            RDREGB EOFSW,R2                   ;<OPTION SWITCHES TO R2
00A2D 05053 F0A31  5055            ADD DB,Q,MSK2,R3 & NOP $+D#4      ;<JUMP LOC. ACCORDING...
00A2E 321FE FFFFF  5056            QLDREG IREG                       ;<TO RECORD LENGTH
00A2F 37353 F0797  5058            RDMSKB R3,797                     ;$ 1/4 WORDS PER TRACK AT 116
00A30 0534A 6F4C8  5062            ADD DB,B,MSK1,R10 & JIR POINT3    ;<FUJI. 337 DRIVE TYPE POINTER
00A31 37341 5F800  5068            RDREGB MSK1,R1 & JAR D#0          ;INVALID RECORD LENGTH
```

```
 Addr            Line CDC O.S. - DRIVE SELECTION

00A32 37341 5F843  5070            RDREGB MSK1,R1 & JAR D#67          ;$FUJI. 337 MAX. RECORD (116 WORDS)
00A33 37341 5F800  5071            RDREGB MSK1,R1 & JAR 0             ;$INVALID RECORD LENGTH
00A34 37341 5F800  5072            RDREGB MSK1,R1 & JAR 0             ;$INVALID RECORD LENGTH
                   5090    ;
                   5091    ;(4) FUJITSU SWALLOW-5 DRIVE (M2344KZ)
                   5095            JSR.P REVJK                        ;Check max. useable cyl.
00A35 E0403 1F740  5095 +          SETPGE REVJK_SHR_H#B & JSRP REVJK
00A36 37350 F02E7  5097            RDMSKB R0,MAXCYL4                  ;SWALLOW-5 MAX. DRIVE CYL.=743
00A37 37355 F001A  5098            RDMSKB R5,D#26                     ;SWALLOW-5 MAX. HEAD = 26
00A38 37357 F0400  5100            RDMSKB   R7,VALSECT                ;Swallow-5 Max. sector (hard)
00A39 FFFFF FFFFF  5105            NOP
00A3A 05053 F0A3E  5106            ADD DB,Q,MSK2,R3 & NOP $+D#4       ;<JUMP LOC. ACCORDING...
00A3B 321FE FFFFF  5107            QLDREG IREG                        ;<TO RECORD LENGTH
00A3C 37353 F0910  5109            RDMSKB R3,910                      ;$ 1/4 Words per Track (116 WORDS)
00A3D 0534A 6F0DA  5113            ADD DB,B,MSK1,R10 & JIR POINT4     ;<SWALLOW-5 DRIVE TYPE POINTER
00A3E 37341 5F800  5119            RDREGB MSK1,R1 & JAR D#0           ;INVALID RECORD LENGTH
00A3F 37341 5F850  5121            RDREGB MSK1,R1 & JAR D#80          ;$SWALLOW-5 MAX. RECORD (116 WORDS)
00A40 37341 5F800  5122            RDREGB MSK1,R1 & JAR 0             ;$INVALID RECORD LENGTH
00A41 37341 5F800  5123            RDREGB MSK1,R1 & JAR 0             ;$INVALID RECORD LENGTH
                   5141    ;
                   5142    ;(5) FUJITSU SWALLOW-4 DRIVE
00A42 37354 F026D  5143            RDMSKB R4,D#621                    ;SWALLOW MAX. USABLE CYL.=621
00A43 37350 F026F  5144            RDMSKB R0,MAXCYL5                  ;SWALLOW MAX. DRIVE CYL.=623
00A44 37355 F001A  5145            RDMSKB R5,D#26                     ;SWALLOW MAX. HEAD = 26
00A45 37357 F0400  5146            RDMSKB   R7,VALSECT                ;Swallow Max. Sector (hard)
00A46 FFFFF FFFFF  5148            NOP
00A47 05053 F0A4B  5149            ADD DB,Q,MSK2,R3 & NOP $+D#4       ;<JUMP LOC. ACCORDING...
00A48 321FE FFFFF  5150            QLDREG IREG                        ;<TO RECORD LENGTH
```

Addr Line CDC O.S. - DRIVE SELECTION

```
00A49 37353 F0797  5152            RDMSKB R3,797                    ;$ 1/4 WORDS PER TRACK AT 116
00A4A 0534A 6F012  5156            ADD DB,B,MSK1,R10 & JIR POINT5   ;<SWALLOW-4 DRIVE TYPE POINTER
00A4B 37341 5F800  5162            RDREGB MSK1,R1 & JAR D#0         ;INVALID RECORD LENGTH
00A4C 37341 5F843  5164            RDREGB MSK1,R1 & JAR D#67        ;$SWALLOW-4 MAX. RECORD (116 WORDS)
00A4D 37341 5F800  5165            RDREGB MSK1,R1 & JAR 0           ;$INVALID RECORD LENGTH
00A4E 37341 5F800  5166            RDREGB MSK1,R1 & JAR 0           ;$INVALID RECORD LENGTH
                   5184    ;
                   5185    ;(6) CDC 368-MB. DRIVE (PA8A2)
00A4F 37354 F0961  5186            RDMSKB R4,D#2401                 ;CDC-368 MAX. USABLE CYL.= 2401
00A50 37350 F0981  5187            RDMSKB R0,MAXCYL6                ;CDC-368 MAX. DRIVE CYL.= 2433
00A51 37355 F0009  5188            RDMSKB R5,D#9                    ;CDC-368 MAX. HEAD = 9
00A52 37357 F043F  5189            RDMSKB  R7,VALSECT+D#63          ;CDC-368 Max. Sector = 63
00A53 37372 FFFFF  5191            RDREGB EOFSW,R2                  ;<OPTION SWITCHES TO R2
00A54 05053 F0A58  5192            ADD DB,Q,MSK2,R3 & NOP $+D#4     ;<JUMP LOC. ACCORDING...
00A55 321FE FFFFF  5193            QLDREG IREG                      ;<TO RECORD LENGTH
00A56 37353 F05AA  5195            RDMSKB R3,5AA                    ;$ 1/4 Words per Track at 116
00A57 0534A 6F436  5199            ADD DB,B,MSK1,R10 & JIR POINT6   ;<CDC-368 DRIVE TYPE POINTER
00A58 37341 5F800  5205            RDREGB MSK1,R1 & JAR D#0         ;INVALID RECORD LENGTH
00A59 37341 5F832  5207            RDREGB MSK1,R1 & JAR D#50        ;$CDC-368 MAX. RECORD (116 WORDS)
00A5A 37341 5F800  5208            RDREGB MSK1,R1 & JAR 0           ;$INVALID RECORD LENGTH
00A5B 37341 5F800  5209            RDREGB MSK1,R1 & JAR 0           ;$INVALID RECORD LENGTH
                   5227    ;
                   5228    ;(7) CDC-368 (PA8A2) WITH DEFECT INFORMATION
00A5C 37354 F0961  5229            RDMSKB R4,D#2401                 ;CDC-368 MAX. USABLE CYL.= 2401
00A5D 37350 F097F  5230            RDMSKB R0,MAXCYL7                ;CDC-368 MAX. DRIVE CYL.= 2431
00A5E 37355 F0009  5231            RDMSKB R5,D#9                    ;CDC-368 MAX. HEAD = 9
00A5F 37357 F043F  5232            RDMSKB  R7,VALSECT+D#63          ;CDC-368 Max. Sector = 63
00A60 37372 FFFFF  5234            RDREGB EOFSW,R2                  ;<OPTION SWITCHES TO R2
00A61 05053 F0A65  5235            ADD DB,Q,MSK2,R3 & NOP $+D#4     ;<JUMP LOC. ACCORDING...
00A62 321FE FFFFF  5236            QLDREG IREG                      ;<TO RECORD LENGTH
00A63 37353 F05AA  5238            RDMSKB R3,5AA                    ;$ 1/4 Words per Track at 116
```

Addr Line CDC O.S. -- DRIVE SELECTION

```
00A64 0534A 6F436  5242            ADD DB,B,MSK1,R10 & JIR POINT7     ;<CDC-368 DRIVE TYPE POINTER
00A65 37341 5F800  5248            RDREGB MSK1,R1 & JAR D#0           ;INVALID RECORD LENGTH
00A66 37341 5F832  5250            RDREGB MSK1,R1 & JAR D#50          ;$CDC-368 MAX. RECORD (116 WORDS)
00A67 37341 5F800  5251            RDREGB MSK1,R1 & JAR 0             ;$INVALID RECORD LENGTH
00A68 37341 5F800  5252            RDREGB MSK1,R1 & JAR 0             ;$INVALID RECORD LENGTH
                   5270
00A69 45052 F0800  5271    SEL14:  TSTRBIT R2,800                     ;DISK ARRAY OPTION?
00A6A FFFFF 3326E  5272            BOZ SEL14A                         ;IF NOT, JUMP
00A6B 335F0 FFFFF  5273            OR ZB,RSBZ,,R0                     ;1/2 MAX. DRIVE CYLINDER
00A6C 335F4 FFFFF  5274            OR ZB,RSBZ,,R4                     ;1/2 MAX. USEABLE CYLINDER
00A6D 5535A F0400  5279            RSTRBIT R10,TWOFER                 ;RESET TWO DRIVES FLAG
                   5280    ;
00A6E 0C14F F039C  5281    SEL14A: ADD AZC,FOUTL,R4,RAME & NOP FRSTALT ;SAVE FIRST ALTERNATE CYL.
00A6F 370FF F00BA  5282            RDRAMQ ILDFLG                      ;GET ILD FLAGS
00A70 4605F F0004  5283            TSTQBIT 4                          ;ILD RUNNING?
00A71 FFFFF 33275  5284            BOZ SEL14B                         ;IF NOT, JUMP
00A72 3414F F020A  5285            ALDRAM R4,MAXCYLI                  ;ILD MAX. USEABLE CYLINDER
00A73 3410F F0245  5286            ALDRAM R0,MAXCYLDI                 ;ILD MAX. DRIVE CYLINDER
00A74 3415F F020D  5287            ALDRAM R5,MAXHDI                   ;ILD MAX. HEAD
                   5288    ;
00A75 3414F F02D3  5289    SEL14B: ALDRAM R4,MAXCYLA                  ;SAVE MAX. USABLE CYLINDER
00A76 3410F F02D4  5290            ALDRAM R0,MAXCYLD                  ;SAVE MAX. DRIVE CYLINDER
00A77 3415F F02D5  5291            ALDRAM R5,MAXHDA                   ;SAVE MAX. AMPERIF HEAD
                   5292    ;
00A78 65157 F0400  5293            TSTREQU  R7,VALSECT                ;Valid Sector bit only?
00A79 FFFFF 33A7C  5294            BNZ      $+3                       ;No, jump
00A7A 14317 FFFFF  5295            SBA      AZ,B,R1,R7                ;Max. rec.- 1 = Max. sector
00A7B 35357 F0400  5296            SETRBIT  R7,VALSECT                ;Include the Valid bit
                   5297            JSR.P    LOGICAL                   ;Logical drive no. to Q
00A7C E0400 1F1F9  5297 +          SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
00A7D 0615E F0494  5298            ADD DQ,FOUTL,MSK2,IREG & NOP MAXSECT ;Loc. of Max. sector this drive
00A7E 3417D FFFFF  5299            ALDREG   R7,RAMI                   ;Save Max. sector this drive
                   5300    ;
00A7F 3411F F02D6  5323            ALDRAM R1,AMPREC                   ;SAVE MAX. AMPERIF RECORD
                   5328    ;
00A80 FFFFF 1F342  5329            JSR FLAGLOC                        ;LOC. OF FLAGS THIS DRIVE
```

Addr Line CDC O.S. - DRIVE SELECTION

```
00A81 4505A F0400  5330            TSTRBIT R10,TWOFER                ;TWO DRIVES FOR THIS UNIT?
00A82 603AA 33285  5331            XOR AQ,B,R10,R10 & BUZ $+3        ;IF NOT, JUMP
00A83 0F10F F02F4  5332            ADD DZC,FOUTL,ZERO,RAME & NOP TWODRIV ;SET TWO DRIVES FLAG
00A84 35358 F0400  5333            SETRBIT R8,TWOFER                 ;SET TWO DRIVES BIT
00A85 3418D FFFFF  5341            ALDREG R8,RAMI                    ;SAVE FLAGS FOR THIS DRIVE
00A86 3418F F03E5  5342            ALDRAM R8,LASTFLG                 ;SAME FOR LAST FLAGS
                   5343    ;
00A87 918AF FFFFF  5344            WRTL HSSM,LDRTPTR,R10             ;LOAD DRIVE TYPE POINTER
00A88 9198F FFFFF  5345            WRTL HSSM,LDDRFLG,R8              ;LOAD THE HEADER FLAGS
                   5346    ;
00A89 375F2 F02CD  5347            OR DZ,RSBZ,RAME,R2 & NOP RECLNTH  ;1/2 REC. LENGTH DIVIDE BY 2 =...
00A8A 335F2 1F0CF  5348            OR ZB,RSBZ,,R2 & JSR MULTPLYA     ;1/4 REC. LENGTH X AMPREC =...
00A8B 3410F F02D9  5349            ALDRAM R0,WRDSTRK                 ;1/4 WORDS PER TRACK
                   5350    ;
00A8C 0F3F1 F02D5  5351            ADD DZC,B,RAME,R1 & NOP MAXHDA    ;AMPERIF MAX. HEAD +1...
00A8D 34302 1F0CF  5352            RTOR R0,R2 & JSR MULTPLYA         ; X 1/4 WORDS PER TRACK =...
00A8E 3490F F02D7  5353            ALDRAMU R0,CYLDIVAM               ;1/4 AMPERIF...
00A8F 3410F F02D8  5354            ALDRAM R0,CYLDIVAL                ;CYLINDER DIVISOR
                   5368    ;
                   5369    SEL14C: TSTMBIT 4,ILDFLG                 ;ILD Running?
00A90 373F0 F00BA  5369 +          RDRAMB  R0,ILDFLG
00A91 45150 F0004  5369 +          TSTBIT  R0,4
00A92 FFFFF 33A9B  5370            BNZ     SELRTN                    ;Yes, no need for Max. Address
                   5371    ;
00A93 0C341 FFFFF  5372            ADD AZC,B,R4,R1                   ;MAX. USEABLE AMPERIF CYL.+1 TO R1
00A94 370FF F02D7  5373            RDRAMQ CYLDIVAM                   ;GET AMPERIF...
00A95 D0FF2 FFFFF  5374            RDXFR SMC,,R2                     ;CYLINDER DIVISOR...
00A96 357F2 F02D8  5375            OR DB,LSBZ,RAME,R2 & NOP CYLDIVAL ;INTO R2
00A97 337F2 1F0CF  5376            OR ZB,LSBZ,,R2 & JSR MULTPLYA     ;COMPUTE AND SAVE...
00A98 321FF F02DF  5377            QLDRAM LIMIT1                     ;MAXIMUM...
00A99 3490F F02E0  5378            ALDRAMU R0,LIMIT2                 ;RELATIVE ADDRESS...
00A9A 3410F F02E1  5379            ALDRAM R0,LIMIT3                  ;LIMIT (DISPLAY 4F5)
00A9B 373F7 F024A  5380    SELRTN: RDRAMB  R7,SAVXFRU                ;Get saved R7-upper bits
00A9C D0FF7 FFFFF  5381            RDXFR   SMC,,R7                   ;Move them to R7-upper
00A9D 353F7 AF24B  5382            OR      DB,B,RAME,R7 & RTS SAVXFRL ;Merge R7-lower bits & return
                   5383    ;
                   5384    ;    THIS ROUTINE ATTEMPTS TO READ THE HEADER FROM THE DRIVE.
                   5385    ;
                   5386    SEL15:  JSR.P SEEK31                      ;Check for normal drive status
00A9E E0400 1F192  5386 +          SETPGE SEEK31_SHR_H#B & JSRP SEEK31
00A9F B1800 F0000  5387            WRTE HSSM,LDRTPTR & NOP 0         ;Drive type pointer = 0
00AA0 3710F F0018  5388            RESETRAM SEN415                   ;RESET ERROR RETRY COUNT
00AA1 37350 F0AA3  5389            RDMSKB R0,$+2                     ;ERROR RETRY LOCATION
                   5390            JSR.P SAVSMRTN                    ;SAVE IT IN CASE OF ERROR
00AA2 E0400 1F4C6  5390 +          SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
                   5391    ;
                   5392            JSR.P CHKBUSY                     ;CHECK INTERFACE BUSY
00AA3 E0400 1F535  5392 +          SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
```

Addr Line CDC O.S. - DRIVE SELECTION

```
00AA4 E1000 FFFFF  5393            CTL HSSM,DIFRST                  ;DISK INTERFACE RESET
00AA5 D1F00 F0030  5394            WRTE HSSM,LDXMCJ & NOP RDHDR
00AA6 E100D FFFFF  5395            CTL HSSM,XMCJCMD                 ;INITIATE READ HEADER OPERATION
00AA7 37350 F0AB8  5396            RDMSKB R0,SEL16                  ;SM-DONE RETURN LOC.
                   5397            JSR.P SAVIRPT                    ;SAVE IT
00AA8 E0400 1F4C0  5397 +          SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
00AA9 37350 F61A8  5398            RDMSKB R0,61A8                   ;SET 50-MS. TIMER
                   5399            JSR.P SAVSMTIM                   ;SAVE IT
00AAA E0400 1F4C2  5399 +          SETPGE SAVSMTIM_SHR_H#B & JSRP SAVSMTIM
                   5400            SETMBIT 2,SEN216                 ;Sense 16, Format 2 = Read Headers
00AAB 37350 F0002  5400 +          RDMSKB   R0,2
00AAC 3410F F0025  5400 +          ALDRAM   R0,SEN216
00AAD 3535F F0001  5401            SETRBIT R15,1                    ;SET HEADER READ FLAG
00AAE 37355 F0100  5402            RDMSKB R5,100                    ;SET SM-READ ACTIVE FLAG
                   5403    ;
                   5404    SEL15A: JSR.P SMIRPT                    ;CHECK FOR AN INTERRUPT
00AAF E0400 1F37C  5404 +          SETPGE SMIRPT_SHR_H#B & JSRP SMIRPT
                   5405            DECSMB SMTIME                    ;Decr. SM-timer
00AB0 373F0 F02DC  5405 +          RDRAMB   R0,SMTIME
00AB1 1410F F02DC  5405 +          SBA      AZ,FOUTL,R0,RAME & NOP SMTIME
00AB2 FFFFF 33AAF  5406            BNZ SEL15A                       ;IF NO TIME-OUT, CHECK AGAIN
00AB3 373F0 F02E2  5407            RDRAMB R0,SMTIMEA                ;GET TIMER MSB'S
00AB4 FFFFF 332C9  5408            BOZ SEL17                        ;IF 0, TIME-OUT ERROR
00AB5 1410F F02E2  5409            SBA AZ,FOUTL,R0,RAME & NOP SMTIMEA ;DECR. TIMER MSB'S
00AB6 7710F F02DC  5410            XNOR DZ,FOUTL,ZERO,RAME & NOP SMTIME ;PRESET TIMER LSB'S
00AB7 FFFFF E0AAF  5411            JMP SEL15A                       ;CHECK AGAIN FOR INTERRUPT
                   5412    ;
00AB8 E1000 1F342  5413    SEL16:  CTL HSSM,DIFRST & JSR FLAGLOC    ;LOC. OF HEADER FLAGS
                   5414            JSR.P HSWAITR                    ;WAIT LOOP ROUTINE
00AB9 E0400 1F525  5414 +          SETPGE HSWAITR_SHR_H#B & JSRP HSWAITR
00ABA 3710F F00A6  5415            RESETRAM PARCNT                  ;RESET PARITY RETRY COUNTER
00ABB 5535E F0200  5416            RSTRBIT R14,200                  ;RESET ALT. TRACK FLAG
00ABC 5535F F0401  5417            RSTRBIT R15,401                  ;RESET SM-RETRY & HEADER READ
00ABD 433F2 1F33E  5418            RESETR R2 & JSR SETHED           ;SET HEAD = 0
00ABE E1003 5F2BF  5419            CTL HSSM,MPCRST & LDAR $+1       ;MPC CONTROL RESET
00ABF 37160 4F001  5420            MOVE SMCST,ZERO & PSHLD 1        ;RESET XMC PARITY ERRORS
00AC0 C1088 887FF  5421            RD HSSM,RDDRFLG,R8 & CLPCNT EVCNTZ ;READ THE HEADER FLAGS
                   5422            JSR.P XMCTPAR                    ;CHECK CT-BUS PARITY
00AC1 E0403 1F35B  5422 +          SETPGE XMCTPAR_SHR_H#B & JSRP XMCTPAR
                   5423            BNZ.P HSSMPAR                    ;IF AN ERROR, PROCESS IT
00AC2 E0403 33B62  5423 +          SETPGE HSSMPAR_SHR_H#B & BNZP HSSMPAR
00AC3 55348 C0001  5424            MASK DB,B,MSK1,R8 & LDCT 1       ;DELETE DEFECTIVE FLAG
00AC4 34586 FFFFF  5429            OR AZ,RSBZ,R8,R6
00AC5 45058 F0030  5430            TSTRBIT R8,30                    ;ANY LENGTH BITS?
```

```
 Addr            Line CDC O.S. - DRIVE SELECTION

00AC6 3418D 339ED  5431            ALDREG R8,RAM( & BNZ SEL10B       ;IF YES, CONTINUE; SAVE FLAGS
00AC7 3710F F03E5  5433            RESETRAM LASTFLG
00AC8 FFFFF E0ACB  5434            JMP SEL17A
                   5488    ;
                   5489    ;   UNABLE TO READ DRIVE HEADER
                   5490    ;
                   5491    SEL17:  JSR.P HSWAITR                     ;WAIT LOOP ROUTINE
00AC9 E0400 1F525  5491 +          SETPGE HSWAITR_SHR_H#B & JSRP HSWAITR
00ACA E1016 1F344  5492            CTL HSSM,RSTAG3 & JSR CLRFLAG     ;CLEAR FLAGS FOR THIS DRIVE
00ACB 373F2 F0018  5493    SEL17A: RDRAMB R2,SEN415                  ;GET SM-RETRY COUNT
00ACC FFFFF 1F33E  5494            JSR SETHED                        ;RETRY COUNT IS HEAD NO.
00ACD 45052 F0001  5495            TSTRBIT R2,1                      ;ODD RETRY COUNT?
00ACE FFFFF 33AD0  5496            BNZ $+2                           ;IF YES, JUMP
00ACF B1800 F0080  5497            WRTE HSSM,LDRTPTR & NOP H#80      ;DRIVE POINTER FOR SECTORED DRIVE
00AD0 2D142 F000A  5498            SUBB DB,FOUTL,MSK1,R2 & NOP D#10  ;IS THE COUNT > 10?
                   5499            CJSR.P AGB,SMERR25+1              ;IF NOT, TRY AGAIN
00AD1 E0400 14683  5499 +          SETPGE SMERR25+1_SHR_H#B & CJSRP AGB,SMERR25+1
                   5500    ;THIS JSR IS FOR STACK ADJUSTMENT ONLY.  THERE WILL BE NO RETURN (RTS).
                   5501    ;
00AD2 5535F F0401  5502            RSTRBIT R15,401                   ;RESET SM-RETRY & HEADER READ
00AD3 433F2 1F33E  5503            RESETR R2 & JSR SETHED            ;SET HEAD = 0
                   5504            TSTMBIT 4,ILDFLG                  ;ILD Running?
00AD4 373F0 F00BA  5504 +          RDRAMB  R0,ILDFLG
00AD5 45150 F0004  5504 +          TSTBIT  R0,4
00AD6 FFFFF 332D8  5505            BOZ SEL19                         ;IF NOT, JUMP
                   5506    ;
00AD7 320FF AF7FF  5507            OR Z0,0 & RTS                     ;RETURN; 0 = ILD RUNNING FLAG
                   5508    ;
                   5535
                   5537    TITLE CDC O.S. - DRIVE SELECTION ERRORS
```

Addr Line CDC O.S. - DRIVE SELECTION ERRORS

```
                    5546    ;
                    5547    SEL19:    SETMBIT  1E,SEN07              ;Sense 7 = Unable to determine...
00AD8 37350 F001E   5547 +            RDMSKB   R0,1E
00AD9 3410F F0007   5547 +            ALDRAM   R0,SEN07
                    5548                                           ;Drive type @ Initial selection
00ADA 37353 F000A   5549              RDMSKB R3,D#10               ;RECOVERY ACTION = 10
                    5550              SETMBIT 40,SEN01             ;SENSE 1 = INVALID Track Format
00ADB 37350 F0040   5550 +            RDMSKB   R0,40
00ADC 3410F F0001   5550 +            ALDRAM   R0,SEN01
00ADD 433F0 E0AFE   5551              RESETR R0 & JMP SEL24B       ;SEND EQUIP. CHECK
                    5552    ;
                    5553    ; DRIVE NOT READY; INTERVENTION REQUIRED
                    5554
00ADE FFFFF 1F345   5555    SEL20:    JSR CLRFLAG+1                ;CLEAR HEADER FLAGS THIS DRIVE
00ADF 3710F F0008   5556    SEL20A:   RESETRAM SEN108              ;CLEAR SENSE BYTE 8, FORMAT 1
00AE0 3710F F000A   5557              RESETRAM SEN110              ;AND SENSE BYTE 10, FORMAT 1
00AE1 3415F F000E   5558              ALDRAM R5,SEN114             ;SAVE DRIVE STATUS FOR SENSE
00AE2 4505E F0020   5559              TSTRBIT R14,20               ;SPECIAL FLAG SET?
00AE3 E100B A3FFF   5560              CTL HSSM,RSTUSEL & CRTS NALUZ ;IF YES, RETURN
                    5561    ;
00AE4 373F7 F00BA   5562              RDRAMB R7,ILDFLG             ;READ ILD FLAGS
00AE5 4505F F2000   5563              TSTRBIT R15,2000             ;CHECK POLLING FLAG
00AE6 FFFFF 332EA   5564              BOZ SEL20B                   ;IF NOT SET, JUMP
                    5565    ;
00AE7 0F306 1F429   5566              ADD DZC,B,ZERO,R6 & JSR POLL20 ;SELECT AN HJD
00AE8 45051 F0004   5567              TSTRBIT R1,4                 ;ILD RUNNING?
00AE9 333F7 A3FFF   5568              OR ZB,B,,R7 & CRTS NALUZ     ;IF IT IS, RETURN
                    5569    ;
00AEA 45357 F0004   5570    SEL20B:   MSKRBIT  R7,4
00AEB FFFFF 33AF3   5571              BNZ SEL22                    ;IF ILD RUNNING, INTERVENTION
                    5572              JSR.P CLREF                  ;RESET ANY E.F.
00AEC E0403 1F583   5572 +            SETPGE CLREF_SHR_H#B & JSRP CLREF
                    5576    ;
                    5577    SEL21:    JSR.P SETATTN                ;SET ATTENTION PEND. THIS DRIVE
00AED E0404 1F02B   5577 +            SETPGE SETATTN_SHR_H#B & JSRP SETATTN
00AEE 4505F F2000   5578              TSTRBIT R15,2000             ;CHECK POLLING FLAG
00AEF FFFFF 332F3   5579              BOZ SEL22                    ;IF NOT SET, JUMP
00AF0 3715E F0180   5583              MOVE MSK2,1REG & NOP H#180   ;RECOVERY ACTION = 6
00AF1 FFFFF E0BFF   5584              JMP POLL16                   ;SEND DEVICE END/UNIT CHECK
                    5585    ;
                    5586    SEL22A:   JSR.P DOUBLE                 ;DOUBLE DRIVE NO. IF NEEDED
00AF2 E0403 1F125   5586 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
                    5587    SEL22:    SETMBIT 40,SEN00             ;Sense 0 = Intervention Required
00AF3 37350 F0040   5587 +            RDMSKB   R0,40
00AF4 3410F F0000   5587 +            ALDRAM   R0,SEN00
```

Addr Line CDC O.S. - DRIVE SELECTION ERRORS

```
                5588            JSR.P   LOGICAL                 ;Logical drive no. to Q
00AF5 E0400 1F1F9 5588 +        SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
00AF6 0615E F0494 5589          ADD DQ,FOUTL,MSK2,IREG & NOP MAXSECT ;Loc. of Max. sector this drive
00AF7 3710D FFFFF 5590          MOVE    ZERO,RAM1               ;Clear it
00AF8 37353 F0006 5591          RDMSKB R3,6                     ;RECOVERY ACTION = 6
00AF9 FFFFF E0D3F 5592          JMP END10                       ;SEND UNIT CHECK
                5593
00AFA 37350 F0010 5594  SEL23:  RDMSKB R0,10                    ;SET SENSE: FORMAT 1, MSG. 0
00AFB 3410F F0007 5595  SEL24:  ALDRAM R0,SEN07                 ;SAVE BYTE 7
00AFC 373F0 F0000 5596          RDRAMB R0,SEN00                 ;Get previous Sense 0
00AFD 37353 F0008 5597  SEL24A: RDMSKB R3,8                     ;RECOVERY ACTION = 8
00AFE 35350 F0010 5598  SEL24B: SETRBIT R0,10                   ;ADD EQUIP. CHECK TO SENSE
00AFF 3410F F0000 5599          ALDRAM R0,SEN00                 ;SAVE SENSE 0
00B00 4505E F0020 5600          TSTRBIT R14,20                  ;SPECIAL FLAG SET?
00B01 E4402 A3FFF 5601          CTL HIO,HDRVGTR & CRTS NALUZ    ;IF YES, RETURN
00B02 FFFFF BF53F 5603          JPOP END10                      ;$ELSE, SEND UNIT CHECK
                5613    ;
00B03 37350 F001F 5614  SEL25:  RDMSKB R0,1F                    ;SENSE 7: INITIAL SELECT TIME-OUT
00B04 370FF F0008 5615          RDRAMQ SEN108                   ;GET SENSE BYTE 8
00B05 3605F F0002 5616          SETQBIT 2                       ;ADD BUSY TO SENSE
00B06 321FF F0008 5617          QLDRAM SEN108                   ;RE-WRITE BYTE 8
00B07 FFFFF E0AFB 5618          JMP SEL24
00B08 37350 F0020 5646  SEL28:  RDMSKB R0,20                    ;Sense 7 = Format 2, Msg. 0
00B09 FFFFF E0AFB 5647          JMP SEL24                       ;Now send Equip. Check
                5648    ;
                5649    ; DRIVE FAULT
                5650
00B0A B2200 F0010 5651  SEL30:  WRTE HSSMA,LDDRB0 & NOP H#10    ;BUS OUT FOR FAULT RESET
00B0B E100A C0009 5652          CTL HSSM,SETAG3 & LDCT 9        ;SET TAG TO RESET FAULT
00B0C C1045 9830C 5653          RD HSSM,RDDRST1,R5 & CJPCNT EVCNTZ,$ ;GET DRIVE STATUS
00B0D E1016 FFFFF 5654          CTL HSSM,RSTAG3
00B0E 45155 F0008 5655          TSTBIT R5,FAULT                 ;CHECK FOR FAULT
00B0F 45045 A3001 5656          AND DB,Q,MSK1,R5 & CRTS ALUZ,READY ;IF FAULT CLEARED, RETURN
                5657    ;
00B10 37350 F0020 5658  SEL31:  RDMSKB R0,20                    ;SET SENSE: DEVICE CHECK
00B11 3410F F0010 5659          ALDRAM R0,SEN121                ;SAVE BYTE 21, FORMAT 1
00B12 3415F F000E 5660          ALDRAM R5,SEN114                ;BAD STATUS FOR SENSE
00B13 37353 F0008 5661          RDMSKB R3,D#8                   ;RECOV. ACTION = 8
00B14 4505F F2000 5662          TSTRBIT R15,2000                ;POLLING FLAG SET?
00B15 FFFFF 33BFA 5663          BNZ POLL15                      ;IF YES, SEND DEVICE END/UNIT CHECK
                5664    ;
00B16 45055 F0004 5665          TSTRBIT R5,SEEKERR              ;CHECK FOR SEEK ERROR
00B17 FFFFF 33327 5666          BOZ SEL35                       ;IF NOT SEEK ERROR, JUMP
                5667            JSR.P SKERLOG                   ;UPDATE SEEK ERROR LOG
00B18 E0400 1F1EB 5667 +        SETPGE SKERLOG_SHR_H#B & JSRP SKERLOG
                5668    ;
                5669    SEL32:  RSTMBIT 10,SEN108               ;Reset Seek Comp. in Sense 8
```

```
 Addr            Line CDC O.S. - DRIVE SELECTION ERRORS

00B19 373F0 F0008  5669 +           RDRAMB   R0,SEN108
00B1A 55350 F0010  5669 +           RSTRBIT  R0,10
00B1B 3410F F0008  5669 +           ALDRAM   R0,SEN108
00B1C 37350 F001A  5670             RDMSKB R0,1A                      ;SET SENSE: SEEK ERROR
00B1D 3410F F0007  5671   SEL32A:   ALDRAM R0,SEN07                   ;SAVE BYTE 7
00B1E 37340 4F001  5672             RDREGB MSK1,R0 & PSHLD 1          ;SET SENSE BYTE 0: SEEK ERROR
                   5673   ;
00B1F C1043 887FF  5674             RD HSSM,RDDRST1,R3 & CLPCNT EVCNTZ ;GET DRIVE STATUS
00B20 3413F F000E  5675             ALDRAM R3,SEN114                  ;SAVE DRIVE STATUS FOR SENSE
00B21 45053 F0008  5676             TSTRBIT R3,FAULT                  ;CHECK FOR FAULT
00B22 FFFFF 332FD  5677             BOZ SEL24A                        ;IF NOT, JUMP
                   5678   ;
                   5679             SETNBIT 80,SEN111                 ;Servo status = Perm. Error
00B23 37350 F0080  5679 +           RDMSKB   R0,80
00B24 3410F F000B  5679 +           ALDRAM   R0,SEN111
00B25 0F10F F000A  5680             ADD DZC,FOUTL,ZERO,RAME & NOP SEN110 ;RESET ACCESS READY BIT
00B26 FFFFF E0AFD  5681             JMP SEL24A                        ;EQUIPMENT CHECK
                   5682   ;
00B27 37350 F0019  5683   SEL35:   RDMSKB R0,19                      ;SET SENSE: FILE STATUS NOT
00B28 FFFFF E0AFB  5684             JMP SEL24                         ;AS EXPECTED
                   5691
                   5692
                   5694   TITLE CDC DRIVE SELECTION - SUB-ROUTINES
```

Addr Line CDC DRIVE SELECTION - SUB-ROUTINES

```
                    5703
                    5704    ;   THE FOLLOWING ROUTINE WILL COMPUTE ONE BIT REPRESENTING A DRIVE
                    5705    ;NUMBER.  UPON ENTRY THE 6 LSB'S OF R0 = LOGICAL DRIVE NO.
                    5706    ;   UPON EXIT:  R3 = BIT FOR THIS DRIVE (24-31)
                    5707    ;               R4 = BIT FOR THIS DRIVE (0-23)
                    5708    ;               SMB LOC. 8A = BIT FOR THIS DRIVE (DISPLAY 42E)
                    5709    ;
                    5710    ;   REGISTERS USED:  R0,R3,R4,Q & IREG
                    5711    ;
00B29 45050 F003F   5712    SEL40:    TSTRBIT R0,3F                 ;ISOLATE DRIVE NO. IN Q
00B2A 37350 F0001   5713              RDMSKB R0,1                   ;BIT POSITION FOR DRIVE 0
00B2B 323F3 FFFFF   5714              QTOR R3                       ;MOVE DRIVE NO. TO R3
00B2C 420FF 33330   5715              RESETQ & BOZ $+4              ;RESET Q.  IF DRIVE 0, JUMP
00B2D 133F3 FFFFF   5716              DECRB R3                      ;DECR. DRIVE NO.
00B2E 336F0 33330   5717              OR ZB,LSBQZ,,R0 & BOZ $+2     ;SHIFT BIT; IF DR. NO.= 0, JUMP
00B2F 133F3 E0B2E   5718              DECRB R3 & JMP $-1            ;DECR. DRIVE NO. & JUMP BACK
                    5719    ;
00B30 3714E F008A   5720              MOVE MSK1,IREG & NOP BITDRV   ;LOC. OF BIT FOR THIS DRIVE
00B31 34304 FFFFF   5721              RTOR R0,R4                    ;24 LSB'S TO R4
00B32 323F3 E0044   5722              QTOR R3 & JMP SMBWRT          ;8 MSB'S TO R3 & SAVE THE BIT
                    5723
00B33 4505D F0018   5724    STRING:   TSTRBIT R13,18                ;ISOLATE DRIVE STRING NO.
00B34 327F2 FFFFF   5725              OR ZQ,LSBZ,,R2                ;MOVE IT INTO...
00B35 01722 FFFFF   5726              ADD AB,LSBZ,R2,R2             ;POSITON AND...
00B36 3412F AF004   5727              ALDREG R2,RAME & RTS SEN04    ;SAVE IT FOR SENSE 4; RETURN
                    5728
                    5729    ; COMPUTE PHYSICAL ADDRESS FOR SENSE BYTES
                    5730    ;
                    5731    ;UPON ENTRY:  R13 = DRIVE NO.
                    5732    ;REGISTERS USED:  R0 & Q
                    5733    ;
                    5734    PHYSAD:   JSR.P LOGICAL                 ;LOGICAL DRIVE TO Q
00B37 E0400 1F1F9   5734 +            SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
00B38 4605F F0007   5735              TSTQBIT 7                     ;MASK DRIVE NO. TO 3 BITS...
00B39 66740 F0007   5736              XOR DQ,LSBZ,MSK1,R0 & NOP D#7 ;COMPLEMENT AND SHIFT...
00B3A 01700 FFFFF   5737              ADD AB,LSBZ,R0,R0             ;THEN 3 PLACES
00B3B 353F0 F0004   5738              OR DB,B,RAME,R0 & NOP SEN04   ;MERGE STRING NO. WITH COMP. BITS
00B3C 3010F F0004   5739              OR AQ,FOUTL,R0,RAME & NOP SEN04 ;COMPLETE PHYS. I.D. TO SENSE 4...
00B3D 3010F AF019   5740              OR AQ,FOUTL,R0,RAME & RTS SEN416 ;AND TO SENSE 16, FORMAT 4; RETURN
                    5741
00B3E 9222F 4F004   5742    SETHED:   WRTL HSSMA,LDDRB0,R2 & PSHLD 4 ;LOAD THE HEAD NUMBER
00B3F E1009 887FF   5743              CTL HSSM,SETAG2 & CLPCNT EVCNTZ ;SET THE HEAD
00B40 E1013 FFFFF   5744              CTL HSSM,RSTAG2               ;RESET TAG 2
00B41 B2200 AF000   5745              WRTE HSSMA,LDDRB0 & RTS 0     ;CLEAR BUS-OUT REG. & RETURN
```

```
 Addr            Line CDC DRIVE SELECTION - SUB-ROUTINES


                 5747
                 5748    ;
                 5749    ;    THIS ROUTINE DETERMINES THE SMB LOCATION OF THE FLAGS FOR THIS DRIVE.
                 5750    ;UPON ENTRY:  R13 = DRIVE NO.
                 5751    ;UPON EXIT:  IREG = SMB LOCATION
                 5752    ;Registers used:  Q & IREG.
                 5753    ;
                 5754    FLAGLOC:  JSR.P LOGICAL                     ;ISOLATE THE LOGICAL DRIVE NO.
00B42 E0400 1F1F9 5754 +           SETPGE LOGICAL_SHR_N#B & JSRP LOGICAL
00B43 0614E AF378 5755             ADD DQ,FOUTL,MSK1,IREG & RTS DRFLAGS ;COMPUTE LOC. & RETURN
                 5756    ;
00B44 B1800 F0000 5757    CLRFLAG:  WRTE HSSM,LDRTPTR & NOP 0         ;DRIVE TYPE POINTER = 0
00B45 0F10F F03E5 5758           ADDC DZ,FOUTL,ZERO,RAME & NOP LASTFLG;PRESET LAST DRIVE TYPE FLAG
00B46 FFFFF 1F342 5759             JSR FLAGLOC                       ;LOC. OF DRIVE FLAGS
00B47 3710D AF7FF 5760             MOVE ZERO,RAM1 & RTS              ;CLEAR THEM & RETURN
                 5761
                 5762    ;
                 5763    ;    THIS ROUTINE GETS THE SAVED BIT FOR THIS DRIVE.
                 5764    ;UPON EXIT:  R0 & R3 = BIT FOR THIS DRIVE (24-31) OR...
                 5765    ;            Q & R4  = BIT FOR THIS DRIVE (0-23)
                 5766    ;REGISTERS USED:  R0,R1,R3,R4,Q & IREG.
                 5767    ;
00B48 3714E F008A 5768    GETDRBIT: MOVE MSK1,IREG & NOP BITDRV       ;LOC. OF BIT FOR THIS DRIVE
00B49 37160 1F0CD 5769             MOVE SMCST,ZERO & JSR SMBRDA$      ;GET IT AND...
00B4A 34304 FFFFF 5770             RTOR R0,R4                         ;MOVE IT TO R4 (0-23) OR...
00B4B 323F3 AF7FF 5771             QTOR R3 & RTS                      ;MOVE IT TO R3 (24-31); RETURN
                 5772
                 5773
                 5775    TITLE CDC O.S. - ATTENTION POLLING
```

Addr Line CDC O.S. -- ATTENTION POLLING

```
5784
5785    ;     THE ATTENTION POLLING ROUTINE WILL BE ENTERED FROM THE WAIT LOOP IF IT
5786    ;IS DETERMINED THAT AN ATTENTION INTERRUPT IS PENDING ON ANY DRIVE.
5787    ;EACH TIME THE ROUTINE IS ENTERED ONLY ONE DRIVE WILL BE SELECTED.  IF THIS
5788    ;DRIVE IS ON-LINE & READY "ATTENTION" STATUS WILL BE PRESENTED TO ONE CHANNEL
5789    ;ONLY.  ON SUBSEQUENT ENTRIES "ATTENTION" WILL BE PRESENTED TO ALL OTHER
5790    ;CHANNELS CONNECTED TO THIS CONTROLLER.
5791    ;     IF THE DRIVE IS NOT YET ON-LINE & READY, A MASK BIT WILL BE SET FOR THIS
5792    ;DRIVE SO THAT THE SAME DRIVE WILL NOT BE SELECTED ON THE NEXT ENTRY (UNLESS
5793    ;THERE IS ONLY ONE DRIVE BEING POLLED).  THE PROGRAM WILL RETURN TO THE WAIT
5794    ;LOOP.
5795    ;
5796    ;     THE REGISTERS IN POLLING ARE:
5797    ;
5798    ;       R0,R1,R2 & Q = SCRATCH
5799    ;       R3 = BIT FOR THIS DRIVE (24-31)
5800    ;       R4 = BIT FOR THIS DRIVE (0-23)
5801    ;       R5 = DRIVE STATUS
5802    ;       R6 = MASK BITS (24-31)
5803    ;       R7 = MASK BITS (0-23)
5804    ;       R8 = DRIVE BUSY TIMER (SEEK COMP. POLLING)
5805    ;       R9 = POLLING DELAY TIMER (SEEK COMP. POLLING)
5806    ;       R10,R11 & R12 = NOT USED
5807    ;       R13 = DRIVE NO.
5808    ;       R14 & R15 = FLAG REGISTERS
5809    ;       IREG = ADDRESS OF ATTENTION PENDING & ATTENTION MASK
```

Addr Line CDC O.S. - ATTENTION POLLING

```
                    5811
                    5812
00B4C 0F1EE 1F0CD   5813    POLL00:   INCRD IREG,IREG & JSR SMBRDA       ;GET ATTENTION MASK
00B4D 34307 FFFFF   5814    POLL01:   RTOR R0,R7                        ;MOVE LSB'S TO R7
00B4E 323F6 FFFFF   5815              QTOR R6                           ;MOVE MSB'S TO R6
                    5816    ;
00B4F 3705F F0005   5817              RDMSKQ 5
00B50 2E1EE 1F0CD   5818              SUBB DQ,FOUTL,IREG,IREG & JSR SMBRDA ;GET ATTENTION PENDING
00B51 51370 5F356   5819              MASK AB,B,R7,R0 & LDAR POLL05       ;IF ANY ATTENTION PENDING BITS...
00B52 5006F 33C45   5820              MASK AQ,Q,R6 & BNZ DRVNBR           ;DRIVES 0-23, COMPUTE DRIVE NO.
00B53 FFFFF 33C45   5821              BNZ DRVNBR                          ;DRIVES 24-31, COMPUTE DRIVE NO.
                    5822    ;
00B54 0F1EE 1F0CE   5823              INCRD IREG,IREG & JSR SMBWRTA       ;CLEAR THE ATTENTION MASK
00B55 FFFFF E0B4D   5824              JMP POLL01                          ;AND CHECK AGAIN
                    5825    ;
00B56 373E2 FFFFF   5826    POLL05:   RDREGB IREG,R2                      ;SAVE IREG FOR LATER
00B57 3714E F006F   5827              MOVE MSK1,IREG & NOP BITILD
00B58 FFFFF 1F0CD   5828              JSR SMBRDA                          ;GET BIT FOR THE ILD DRIVE
00B59 3412E FFFFF   5829              ALDREG R2,IREG                      ;RESTORE IREG
00B5A 41340 5F38F   5830              AND AB,B,R4,R0 & LDAR POLL07        ;IF THIS ATTENTION PENDING IS...
00B5B 4003F 33B8A   5831              AND AQ,Q,R3 & BNZ POLL06            ;ON THE CURRENTLY ACTIVE ILD...
00B5C 0F306 33B8A   5832              ADD DZC,B,ZERO,R6 & BNZ POLL06      ;DRIVE, EXIT POLLING; DR. CNTR.=1
                    5833    ;
00B5D E100B FFFFF   5834              CTL HSSM,RSTUSEL                    ;DE-SELECT ANY ILD DRIVE
                    5840              JSR.P DOUBLE                        ;DOUBLE DRIVE NO. IF NEEDED
00B5E E0403 1F125   5840 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
                    5841    ;
00B5F 4505D F003F   5842    POLL05A:  TSTRBIT R13,3F                      ;MASK OUT ANY STATUS
00B60 B2200 F0000   5843              WRTE HSSMA,LDDRBO & NOP 0           ;CLEAR DRIVE BUS OUT REG.
00B61 323F1 5F38D   5844              QTOR R1 & LDAR POLL06A              ;MOVE DRIVE NO. TO R1
00B62 46340 F0030   5845              AND DQ,B,MSK1,R0 & NOP H#30         ;CARD CAGE BITS (CHALLENGER) TO R0
00B63 91E1F 4F00A   5846              WRTL HSSM,LDDRUSEL,R1 & PSHLD 10    ;LOAD UNIT SEL. REG. & DELAY COUNT
00B64 C1045 FFFFF   5847              RD HSSM,RDDRST1,R5
00B65 C1045 B8368   5848              RD HSSM,RDDRST1,R5 & CJPOP EVCNTZ,$+3 ;READ DRIVE STATUS
00B66 E1007 83FFF   5849              CTL HSSM,SETUSEL & CLPCNT NALUZ     ;IF NONE, JUMP BACK; SELECT DRIVE
                    5850    ;
00B67 91E0F FFFFF   5851              WRTL HSSM,LDDRUSEL,R0               ;CARD CAGE SELECT ONLY
00B68 C1045 98368   5852              RD HSSM,RDDRST1,R5 & CJPCNT EVCNTZ,$ ;DELAY WHILE STATUS STABILIZES
00B69 45055 F0001   5853              TSTRBIT R5,READY                    ;UNIT READY?
00B6A FFFFF 3338A   5854              BOZ POLL06                          ;IF NOT, JUMP
                    5855    ;
                    5856              JSR.P GETFLAG                       ;GET THE FLAGS FOR THIS DRIVE
00B6B E0402 1F188   5856 +            SETPGE GETFLAG_SHR_H#B & JSRP GETFLAG
00B6C 45058 F0400   5857              TSTRBIT R8,TWOFER                   ;TWO DRIVES FOR THIS UNIT?
00B6D 333F6 33372   5858              OR ZB,B,,R6 & BOZ POLL05B           ;IF NOT, JUMP
00B6E 133F6 33372   5859              DECRB R6 & BOZ POLL05B              ;DECR. DR. CNTR.; IF 0, JUMP
00B6F FFFFF 1F60F   5860              JSR RELDRIV                         ;RELEASE & DE-SELECT DRIVE
00B70 3535D F0001   5861              SETRBIT R13,1                       ;INCR. DRIVE NO.+1
00B71 FFFFF E085F   5862              JMP POLL05A                         ;CHECK THE SECOND DRIVE
                    5863    ;
```

Addr Line CDC O.S. - ATTENTION POLLING

```
                    5864    POLL05B:  RSTMBIT  4,ILDFLG                ;Reset ILD Running flag
00B72 373F0 F00BA   5864 +            RDRAMB   R0,ILDFLG
00B73 55350 F0004   5864 +            RSTRBIT  R0,4
00B74 3410F F00BA   5864 +            ALDRAM   R0,ILDFLG
00B75 433F1 1F333   5865              RESETR R1 & JSR STRING            ;COMPUTE DRIVE STRING NO.
00B76 433F0 1F337   5866              RESETR R0 & JSR PHYSAD            ;COMPUTE PHYSICAL ADDRESS
00B77 37352 F000A   5867              RDMSKB R2,H#A                     ;SENSE 7: READ LABEL REQUIRED
                    5868
                    5869    POLL05C:  SETMBIT  H#C,SEN10B               ;Sense 8 = On-line/Pack Change
00B78 37350 F000C   5869 +            RDMSKB   R0,H#C
00B79 3410F F0008   5869 +            ALDRAM   R0,SEN10B
00B7A 3415F F000E   5870              ALDRAM R5,SEN114                  ;SAVE BYTE 14, FORMAT 1
00B7B 341DF F000D   5871              ALDRAM R13,SEN113                 ;SAVE DRIVE NO. FOR SENSE
                    5872              SETMBIT  H#E4,SEN115              ;Set Sense 15, Format 1
00B7C 37350 F00E4   5872 +            RDMSKB   R0,H#E4
00B7D 3410F F000F   5872 +            ALDRAM   R0,SEN115
                    5873              SETMBIT  21,SEN110                ;Sense 10 = Access Ready/Head Load
00B7E 37350 F0021   5873 +            RDMSKB   R0,21
00B7F 3410F F000A   5873 +            ALDRAM   R0,SEN110
00B80 3412F F0007   5874              ALDRAM R2,SEN07                   ;SAVE BYTE 7
                    5875    ;
                    5876              JSR.P ERRLOG                      ;COMPUTE LOC. OF ERROR/USAGE LOG.
00B81 E0400 1F350   5876 +            SETPGE ERRLOG_SHR_H#B & JSRP ERRLOG
00B82 3412E C0008   5877              ALDREG R2,IREG & LDCT 8           ;3 WORDS TO CLEAR
                    5878              JSR.P CLR25                       ;CLEAR ERROR/USAGE LOG THIS DRIVE
00B83 E0400 1F33D   5878 +            SETPGE CLR25_SHR_H#B & JSRP CLR25
00B84 4505D F1000   5879              TSTRBIT R13,1000                  ;CHECK FOR BUSY STATUS
00B85 0F70A 33D70   5880              ADD DZC,LSBZ,ZERO,R10 & BNZ END30+1 ;IF BUSY, SEND ATTN./BUSY
00B86 370FF F00BA   5881              RDRAMQ ILDFLG                     ;GET ILD FLAGS
00B87 4605F F0004   5882              TSTQBIT 4                         ;CHECK ILD RUNNING
00B88 FFFFF 13399   5883              CJSR ALUZ,POLL09                  ;IF NOT, SELECT AN HAIO
00B89 FFFFF E0D6F   5884              JMP END30                         ;SEND ATTENTION STATUS
                    5885    ;
00B8A 0F3E2 FFFFF   5886    POLL06:   ADD DZC,B,IREG,R2                 ;LOC. OF ATTENTION MASK
                    5887              JSR.P ADDABIT                     ;ADD THE BIT FOR THIS DRIVE
00B8B E0400 1F1FF   5887 +            SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
00B8C FFFFF 5FFFF   5888              JAR
                    5889    ;
00B8D 45055 F0008   5890    POLL06A:  TSTRBIT R5,FAULT                  ;CHECK FOR FAULT
00B8E FFFFF 33B92   5891              BNZ POLL08                        ;IF FAULT, TRY TO RESET IT
00B8F FFFFF 1F60F   5892    POLL07:   JSR RELDRIV                       ;RELEASE & DE-SELECT DRIVE
00B90 373F2 F002F   5893              RDRAMB R2,ATTNHIO                 ;GET ATTENTION HAIO NO.
00B91 FFFFF E0017   5894              JMP WAIT12                        ;RETURN TO WAIT LOOP
                    5895    ;
00B92 B2200 F0010   5896    POLL08:   WRTE HSSNA,LDDRB0 & NOP H#10      ;LOAD BUS FOR FAULT CLEAR
00B93 E100A 4F00A   5897              CTL HSSN,SETAG3 & PSHLD 10        ;RESET FAULT
00B94 C1045 887FF   5898              RD HSSN,RDDRST1,R5 & CLPCNT EVCNTZ ;READ DRIVE STATUS
00B95 45055 F0008   5899              TSTRBIT R5,FAULT                  ;CHECK FOR FAULT
```

```
 Addr            Line CDC O.S. - ATTENTION POLLING

00B96 E1016 3338F  5900            CTL HSSM,RSTAG3 & BOZ POLL07      ;IF FAULT RESET, NORMAL EXIT
                   5901    ;
00B97 FFFFF 1F399  5902            JSR POLL09                        ;SELECT THE HAIO
00B98 FFFFF E0B10  5903            JMP SEL31                         ;SEND EQUIP. CHECK
                   5904    ;
00B99 370FF F002F  5905    POLL09: RDRAMQ ATTNHIO                    ;GET ATTN. HAIO NO.
00B9A 0615E F0B9C  5906            ADD DQ,FOUTL,MSK2,IREG & NOP $+D#2
00B9B 433F0 6F7FF  5907            RESETR R0 & JIR                   ;SELECT THE PROPER HAIO
                   5908    ;
00B9C E407E E0BA0  5909            CTL HIO,SPILCH1 & JMP $+4         ;SELECT HAIO-1
00B9D E407D E0BA0  5910            CTL HIO,SPILCH2 & JMP $+3         ;SELECT HAIO-2
00B9E E407B E0BA0  5911            CTL HIO,SPILCH3 & JMP $+2         ;SELECT HAIO-3
00B9F E4077 FFFFF  5912            CTL HIO,SPILCH4                   ;SELECT HAIO-4
00BA0 0F301 4F003  5913            ADD DZC,B,ZERO,R1 & PSHLD 3       ;HAIO SELECTED MASK
00BA1 451C1 887FF  5914            AND DB,FOUTL,HSINST,R1 & CLPCNT EVCNTZ ;CHECK HAIO SELECTED
00BA2 FFFFF A3FFF  5915            CRTS NALUZ                        ;IF SELECTED, RETURN
                   5916    ;
00BA3 327F2 FFFFF  5917            OR ZQ,LSBZ,,R2                    ;COMPUTE LOCATION...
00BA4 00722 FFFFF  5918            ADD AQ,LSBZ,R2,R2                 ;OF ATTENTION...
00BA5 3705F F0072  5919            RDMSKQ ATTN1                      ;PENDING BITS...
00BA6 0012E FFFFF  5920            ADD AQ,FOUTL,R2,IREG              ;FOR THIS HAIO
00BA7 420FF 1F0CE  5921            RESETQ & JSR SMBWRTA              ;CLEAR ATTN. PEND. THIS HAIO
                   5922    ;
00BA8 FFFFF 1F60F  5923            JSR RELDRIV                       ;RELEASE & DE-SELECT THE DRIVE
00BA9 FFFFF E0B8F  5924            JMP POLL07                        ;EXIT
                   5925
                   5926
                   5928    TITLE CDC O.S. - SEEK COMPLETE POLLING
                   5936
                   5937    ;     THE SEEK COMPLETE POLLING ROUTINE WILL BE ENTERED FROM THE WAIT LOOP IF
                   5938    ;IT IS DETERMINED THAT A PRE-SEEK IS IN PROCESS ON ANY DRIVE.  UPON EACH ENTRY
                   5939    ;ONLY ONE DRIVE WILL BE SELECTED.  THE STATUS OF THE DRIVE WILL BE CHECKED TO
                   5940    ;DETERMINE IF THE DRIVE IS "ON CYLINDER".  IF ON CYLINDER, "DEVICE END" STATUS
                   5941    ;WILL BE PRESENTED ONLY TO THE CHANNEL WHICH INITIATED THE SEEK TO THIS DRIVE.
                   5942    ;     IF NOT ON CYLINDER, A MASK BIT WILL BE SET SO THAT THE SAME DRIVE WILL NOT
                   5943    ;BE CHECKED ON THE NEXT ENTRY (UNLESS THERE IS ONLY ONE DRIVE WITH AN ACTIVE
                   5944    ;SEEK), THE DRIVE WILL BE DE-SELECTED AND THE PROGRAM WILL RETURN TO THE WAIT
                   5945    ;LOOP.
                   5946    ;     THE PROGRAM WILL ATTEMPT TO CORRECT ANY DRIVE ERRORS ENCOUNTERED WHILE
                   5947    ;POLLING.
                   5948    ;
                   5949    ;     THE REGISTERS ARE THE SAME AS FOR ATTENTION POLLING.
                   5950    ;
```

Addr Line CDC O.S. -- SEEK COMPLETE POLLING

```
                    5952
                    5953
00BAA 2D059 F0020   5954    POLL10:  SUBB DB,0,MSK2,R9 & NOP H#20       ;If Poll delay timer is large...
00BAB E7C00 3541C   5955             CTL OCP,OCPDSEL & CJMP ALB,POLL17E+2 ;De-select OCP & return to Wait
00BAC 3714E F0051   5956             MOVE MSK1,IREG & NOP SKMASK
00BAD 373E5 1F0CD   5957             RDREGB IREG,R5 & JSR SMBRDA        ;GET SEEK COMPLETE MASK
00BAE 34307 FFFFF   5958    POLL11:  RTOR R0,R7                         ;MOVE LSB'S TO R7
00BAF 323F6 FFFFF   5959             QTOR R6                            ;MOVE MSB'S TO R6
                    5960    ;
00BB0 3412E 1F0CD   5961             ALDREG R2,IREG & JSR SMBRDA        ;GET SEEK COMPLETE PENDING BITS
00BB1 51370 5F3BE   5962             MASK AB,B,R7,R0 & LDAR POLL13      ;IF ANY SEEK COMPLETE PENDING...
00BB2 5006F 33C45   5963             MASK AQ,Q,R6 & BNZ DRVNBR          ;DRIVES 0-23, COMPUTE DRIVE NO.
00BB3 FFFFF 33C45   5964             BNZ DRVNBR                         ;DRIVES 24-31, COMPUTE DRIVE NO.
                    5965    ;
00BB4 3415E 1F0CE   5966             ALDREG R5,IREG & JSR SMBWRTA       ;CLEAR SEEK COMPLETE MASK
00BB5 333F9 FFFFF   5967             OR ZB,B,,R9                        ;TIMER EXPIRED?
00BB6 FFFFF 333AE   5968             BOZ POLL11                         ;YES, CHECK AGAIN
00BB7 133F9 E0BAE   5969             DECRB R9 & JMP POLL11              ;ELSE, DECR. TIMER & CHECK AGAIN
                    5970    ;
                    5971    ;     ILD SEEK COMPLETE POLLING
                    5972    ;
00BB8 3714E F0063   5973    POLL12:  MOVE MSK1,IREG & NOP SKILD
00BB9 FFFFF 1F0CD   5974             JSR SMBRDA                         ;GET ILD SEEK COMP. PENDING BIT
00BBA 373F7 F00BA   5975             RDRAMB R7,ILDFLG                   ;GET ILD FLAGS
00BBB 35357 F0004   5976             SETRBIT R7,4                       ;SET ILD RUNNING FLAG
00BBC FFFFF 5F3C0   5978             LDAR     POLL14                    ;$Return loc. from DRVNBR
00BBD 433F9 E0C45   5982             RESETR R9 & JMP DRVNBR             ;COMPUTE DRIVE NUMBER
                    5989    ;
00BBE 373F7 F00BA   5990    POLL13:  RDRAMB R7,ILDFLG                   ;GET THE ILD FLAGS
00BBF 55357 F0004   5991             RSTRBIT R7,4                       ;DELETE ILD RUNNING FLAG
                    5995    ;
00BC0 3417F F00BA   5996    POLL14:  ALDRAM R7,ILDFLG                   ;SAVE ILD FLAGS
00BC1 3535F F2000   5997             SETRBIT R15,2000                   ;SET POLLING FLAG
00BC2 343D0 1F329   5998             RTOR R13,R0 & JSR SEL40            ;SET THE BIT FOR THIS DRIVE
                    5999             JSR.P DOUBLE                       ;DOUBLE DRIVE NO. IF NEEDED
00BC3 E0403 1F125   5999 +           SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
                    6000    ;
00BC4 3714E F02B5   6001             MOVE MSK1,IREG & NOP DRONCYL
00BC5 FFFFF 1F0CD   6002             JSR SMBRDA                         ;GET DRIVES ON CYLINDER
00BC6 41140 333C9   6003             AND AB,FOUTL,R4,R0 & BOZ POLL14D   ;IF NONE, MUST SELECT; CHECK 0-23
00BC7 40130 33C0E   6004             AND AQ,FOUTL,R3,ZERO & BNZ POLL17B ;IF THIS DRIVE (0-23), DON'T SELECT
00BC8 FFFFF 33C0E   6005             BNZ POLL17B                        ;IF THIS DRIVE(24-31), DON'T SELECT
                    6006    ;
                    6007    POLL14D: EQU $
```

Addr Line CDC O.S. -- SEEK COMPLETE POLLING

```
                    6015    POLL14E:  JSR.P GETFLAG                     ;GET FLAGS THIS DRIVE
00BC9 E0402 1F18B   6015 +            SETPGE GETFLAG_SHR_H#B & JSRP GETFLAG
00BCA FFFFF 139E6   6016              CJSR NALUZ,SEL10A+1               ;IF ANY, COMPUTE CONSTANTS
00BCB FFFFF 1F348   6017              JSR GETDRBIT$                     ;GET THE BIT FOR THIS DRIVE
                    6018              JSR.P CYL00                       ;LOC. OF CUR. CYLINDER
00BCC E0400 1F1E9   6018 +            SETPGE CYL00_SHR_H#B & JSRP CYL00
00BCD 5535E F0800   6019              RSTRBIT R14,800                   ;RESET SECOND DRIVE FLAG
00BCE 45158 F0400   6020              TSTBIT R8,TWOFER                  ;TWO DRIVES THIS UNIT?
00BCF 323FD 333D4   6021              QTOR R13 & BOZ POLL14A            ;IF NOT, SELECT THE DRIVE
                    6022    ;
00BD0 375F1 F02D4   6023              OR DZ,RSBZ,RAME,R1 & NOP MAXCYLD  ;1/2 MAX. DRIVE CYL. TO R1
00BD1 2D1D1 FFFFF   6024              SUBB DB,FOUTL,RAMI,R1             ;CUR. CYL.> 1/2 MAX. DRIVE CYL?
00BD2 FFFFF 34BD4   6025              CJMP NAGB,POLL14A                 ;NO, JUMP
00BD3 3535E F0800   6026              SETRBIT R14,800                   ;SET SECOND DRIVE FLAG
                    6027    ;
00BD4 FFFFF 1F187   6028    POLL14A:  JSR SEL01                         ;CHECK DRIVE ON LINE
00BD5 45055 F0046   6029              TSTRBIT R5,ONCYL+SEEKERR+BUSY     ;CHECK ON CYL, SEEK ERR. & BUSY
00BD6 66341 F0002   6030              XOR DQ,B,MSK1,R1 & NOP ONCYL      ;IF ON CYLINDER ONLY...
00BD7 333F9 33407   6031              OR ZB,B,,R9 & BOZ POLL17          ;SEND DEVICE END STATUS
                    6032    ;
00BD8 37352 F0051   6033              RDMSKB R2,SKMASK                  ;ADD THE BIT FOR THIS DRIVE...
                    6034              JSR.P ADDABIT                     ;TO THE SEEK COMP. MASK
00BD9 E0400 1F1FF   6034 +            SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
                    6035    ;
00BDA 45055 F0004   6036              TSTRBIT R5,SEEKERR                ;CHECK SEEK ERROR
00BDB 34347 33BED   6037              RTOR    R4,R7 & BNZ POLL14C+1     ;IF SET, JUMP
                    6038    POLL14B:  EQU $
00BDC 373F1 F00BA   6042              RDRAMB R1,ILDFLG
00BDD 45051 F0004   6043              TSTRBIT R1,4                      ;ILD RUNNING?
                    6044              BNZ.P WAIT00                      ;IF YES, START WAIT LOOP
00BDE E0400 33B02   6044 +            SETPGE WAIT00_SHR_H#B & BNZP WAIT00
00BDF E100B FFFFF   6045              CTL HSSM,RSTUSEL                  ;DE-SELECT THE DRIVE
00BE0 1D349 F0003   6046              SUBA DB,B,MSK1,R9 & NOP D#3       ;DECR. DEV. END DELAY TIMER
00BE1 333F9 31BE3   6047              OR ZB,B,,R9 & CJMP NCARRY,$+2     ;IF NEGATIVE NO., JUMP
                    6048              BNZ.P   WAIT02                    ;If not 0, return to Wait Loop
00BE2 E0400 33809   6048 +            SETPGE WAIT02_SHR_H#B & BNZP WAIT02
                    6049    ;
00BE3 3715E F02B5   6050              MOVE    MSK2,IREG & NOP DRONCYL   ;Loc. of Drive On-cyl. bits
00BE4 433F9 1F0CD   6051              RESETR  R9 & JSR SMBRDA           ;Any drives on-cylinder?
00BE5 34306 333EB   6052              RTOR    R0,R6 & BOZ POLL14X       ;If not, return to the Wait Loop
                    6053    ;
00BE6 3715E F03FC   6054              MOVE    MSK2,IREG & NOP RPSPEND   ;Loc. of R.P.S. Pending drives
00BE7 323F2 1F0CD   6055              QTOR    R2 & JSR SMBRDA           ;Any R.P.S. seeks pending?
00BE8 41160 333EB   6056              AND AB,FOUTL,R6,R0 & BOZ POLL14X  ;No, Wait Loop; Drive 0-23?
00BE9 4012F 33C15   6057              AND AQ,FOUTL,R2 & BNZ POLL17D     ;Yes, R.P.S. Polling; Drive 24-31?
00BEA FFFFF 33C15   6058              BNZ     POLL17D                   ;Yes, R.P.S. Polling
00BEB FFFFF E0009   6059    POLL14X:  JMP     WAIT02                    ;Continue Wait Loop
                    6060    ;
```

Addr Line CDC O.S. - SEEK COMPLETE POLLING

```
                6061    ; SEEK ERROR WHILE POLLING
                6062    ;
00DEC 34374 FFFFF  6063    POLL14C:  RTOR R7,R4                        ;RESTORE BIT FOR THIS DRIVE (0-23)
00DED 3415F F000E  6064              ALDRAM R5,SEN114                  ;BAD DRIVE STATUS FOR SENSE
00DEE 37352 F008D  6065              RDMSKB R2,SKERRBIT                ;ADD THE BIT FOR THIS DRIVE...
                6066              JSR.P ADDABIT                     ;TO THE SEEK ERROR BITS
00DEF E0400 1F1FF  6066 +            SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
00DF0 373F5 F002C  6067              RDRAMB R5,SKRETRY                 ;GET SEEK RETRY COUNT
                6068              CJSR.P ALUZ,SKERLOG               ;IF COUNT=0, INCR. SEEK ERROR LOG
00DF1 E0400 131EB  6068 +            SETPGE SKERLOG_SHR_H#B & CJSRP ALUZ,SKERLOG
                6069    ;
00DF2 FFFFF 39BFA  6070              CJMP    LOG,POLL15                ;If forced logging, error
00DF3 45055 F0008  6071              TSTRBIT R5,8                      ;IF 8 RE-TRIES...
00DF4 FFFFF 33BFA  6072              BNZ POLL15                        ;PERMANENT SEEK ERROR
                6073    ;
00DF5 0C15F F002C  6074              ADD AZC,FOUTL,R5,RAME & NOP SKRETRY ;INCR. RETRY COUNT
00DF6 3535F F0080  6075              SETRBIT R15,80                    ;SET SEEK ONLY FLAG
                6076    ;
00DF7 0F10F F0302  6077              ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag
                6078              JSR.P CYL00                       ;LOC. OF CUR. CYL. THIS DRIVE
00DF8 E0400 1F1E9  6078 +            SETPGE CYL00_SHR_H#B & JSRP CYL00
00DF9 373D8 E0186  6079              RDREGB RAMI,R11 & JMP SEEK22      ;RE-ZERO & RE-SEEK THE DRIVE
                6080    ;
                6081    ;   Permanent Polling Seek Error.
                6082    ;
00DFA 3710F F002C  6083    POLL15:   RESETRAM SKRETRY                  ;RESET RETRY COUNT
00DFB 0F306 1F429  6084              ADD DZC,B,ZERO,R6 & JSR POLL20    ;SELECT THE HAIO
00DFC 3715E F0200  6085              MOVE MSK2,IREG & NOP H#200        ;RECOVERY ACTION = 8
                6086    ;
                6087              SETMBIT 80,SEN01                  ;Sense 1 = Permanent Error
00DFD 37350 F0080  6087 +            RDMSKB   R0,80
00DFE 3410F F0001  6087 +            ALDRAM   R0,SEN01
                6088    ;
00DFF 3535D F0600  6089    POLL16:   SETRBIT R13,600                   ;SET STATUS: DEVICE END/UNIT CHECK
00C00 37359 F0005  6090              RDMSKB R9,5                       ;STATUS ACTION = 5
                6095    ;
00C01 373F1 F00BA  6096              RDRAMB R1,ILDFLG                  ;GET ILD FLAGS
00C02 45051 F0004  6097              TSTRBIT R1,4                      ;CHECK FOR ILD RUNNING
00C03 3439A 33426  6098              RTOR R9,R10 & BOZ POLL19          ;IF NOT, SEND UNIT CHECK
00C04 55351 F0002  6099              RSTRBIT R1,2                      ;RESET ILD SEEK COMP. PEND. FLAG
00C05 3411F F00BA  6100              ALDRAM R1,ILDFLG
00C06 FFFFF E0B19  6101              JMP SEL32                         ;NOW SEND EQUIP. CHECK
                6102    ;
00C07 FFFFF 3340E  6103    POLL17:   BOZ POLL17B                       ;IF TIMER EXPIRED, JUMP
00C08 37352 F02B5  6104              RDMSKB R2,DRUNCYL                 ;LOC. OF DRIVE ON CYL. BITS
                6105              JSR.P ADDABIT                     ;ADD THE BIT FOR THIS DRIVE
00C09 E0400 1F1FF  6105 +            SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
                6106    ;
```

```
 Addr            Line CDC O.S. - SEEK COMPLETE POLLING

00C0A 1D349 F0003  6107            SUBA DB,B,MSK1,R9 & NOP M#3       ;DECR. DEV. END DELAY TIMER
00C0B FFFFF 31C0D  6108            CJMP NCARRY,POLL17A               ;IF NEGATIVE NO., JUMP
00C0C 333F9 E0C19  6110            OR      ZB,B,,R9 & JMP POLL17D1   ;$
                   6115    ;
00C0D 433F9 FFFFF  6116    POLL17A: RESETR R9                         ;CLEAR DEV. END DELAY TIMER
                   6120    ;
00C0E 3714E F008D  6121    POLL17B: MOVE MSK1,IREG & NOP SKERRBIT     ;LOC. OF SEEK ERROR BITS
00C0F FFFFF 1F0CD  6122            JSR SMBRDA                        ;GET SEEK ERROR BITS
00C10 41140 33414  6123            AND AB,FOUTL,R4,R0 & BOZ POLL17C  ;IF NONE, JUMP
00C11 40130 33C13  6124            AND AQ,FOUTL,R3,ZERO & BNZ $+2    ;THIS DRIVE (0-23), JUMP
00C12 FFFFF 33414  6125            BOZ POLL17C                       ;IF NOT DRIVE 24-31, JUMP
00C13 FFFFF E1F63  6126            JMP RESEEK                        ;RESEEK THE DRIVE
                   6127    ;
00C14 34347 FFFFF  6128    POLL17C: RTOR    R4,R7
00C15 333F9 FFFFF  6146    POLL17D: OR ZB,B,,R9
00C16 34374 33C18  6147            RTOR R7,R4 & BNZ $+2              ;IF TIMER NOT EXPIRED, JUMP
00C17 FFFFF E1C24  6148            JMP SECPOL                        ;ANGULAR POSITION POLLING
                   6149    ;
00C18 133F9 FFFFF  6150            DECRB R9                          ;DECR. TIMER
00C19 FFFFF 3340D  6151    POLL17D1: BOZ POLL17A                      ;IF ZERO, SEND DEVICE END
                   6153    ;
00C1A 37352 F0051  6161    POLL17E: RDMSKB R2,SKMASK                  ;DON'T POLL THIS DRIVE THE...
                   6162            JSR.P ADDABIT                     ;NEXT TIME THRU POLLING
00C1B E0400 1F1FF  6162 +          SETPGE ADDABIT_SHR_M#B & JSRP ADDABIT
00C1C E100B E0009  6163            CTL HSSM,RSTUSEL & JMP WAIT02     ;RETURN TO THE WAIT LOOP
                   6164    ;
00C1D 3709F FFFFF  6165    POLL18:  RDREGQ SPIST                      ;ANY E.F?
                   6166            BNZ.P WAIT03A                     ;YES, PROCESS IT
00C1E E0400 3380E  6166 +          SETPGE WAIT03A_SHR_M#B & BNZP WAIT03A
                   6167            SETMBIT 18,SEN108                 ;Set status: Seek Comp./On Line
00C1F 37350 F0018  6167 +          RDMSKB   R0,18
00C20 3410F F0008  6167 +          ALDRAM   R0,SEN108
                   6168            SETMBIT M#E4,SEN115               ;Set Sense 15, Format 1
00C21 37350 F00E4  6168 +          RDMSKB   R0,M#E4
00C22 3410F F000F  6168 +          ALDRAM   R0,SEN115
00C23 3710F F002C  6169            RESETRAM SKRETRY                  ;RESET RETRY COUNT
00C24 0F306 1F429  6177            ADD DZC,B,ZERO,R6 & JSR POLL20    ;SELECT AN HAIU
00C25 433F5 E0D73  6178            RESETR   R5 & JMP END40           ;Send Device End
                   6179    ;
```

Addr Line CDC O.S. -- SEEK COMPLETE POLLING

```
00C26 371EE FFFFF  6180   POLL19:  MOVE IREG,IREG                    ;STROBE XFER. REG.
00C27 D0FF0 FFFFF  6181            RDXFR SMC,,R0                     ;RECOVERY ACTION TO R0-UPPER
00C28 3130D E0D75  6182            OR AB,B,R0,R13 & JMP END40A       ;SEND DEVICE END/UNIT CHECK
                   6183   ;
00C29 373F1 F00BA  6184   POLL20:  RDRAMB R1,ILDFLG                  ;GET ILD FLAGS
00C2A 45051 F0004  6185            TSTRBIT R1,4                      ;CHECK FOR ILD RUNNING
00C2B 0F302 A3FFF  6186            ADD DZC,B,ZERO,R2 & CRTS NALUZ    ;IF ILD RUNNING, RETURN
                   6190   ;
00C2C 3705F F0C2E  6191   POLL21:  RDMSKQ $+2                        ;COMPUTE SEEK COMPLETE...
00C2D 0016E 5F434  6192            ADD AQ,FOUTL,R6,IREG & LDAR POLL22 ;HAIO NO.
00C2E FFFFF 6F7FF  6193            JIR
                   6194   ;
00C2F 3714E 5F857  6195            MOVE MSK1,IREG & JAR SKHIO1       ;SEEK COMP. HAIO-1 LOC.
00C30 3714E 5F85A  6196            MOVE MSK1,IREG & JAR H#5A         ;SEEK COMP. HAIO-2 LOC.
00C31 3714E 5F85D  6197            MOVE MSK1,IREG & JAR H#5D         ;SEEK COMP. HAIO-3 LOC.
00C32 3714E 5F860  6198            MOVE MSK1,IREG & JAR H#60         ;SEEK COMP. HAIO-4 LOC.
00C33 FFFFF E0C42  6199            JMP POLL23                        ;ERROR, JUMP
```

Addr Line CDC O.S. - SEEK COMPLETE POLLING

```
                6201    ;   Polling HAIO Selection.
                6202    ;
00C34 FFFFF 1F0CD  6203    POLL22:   JSR SnBRDA                    ;GET SEEK COMP. BITS THIS HAIO
00C35 41140 33438  6204              AND AB,FOUTL,R4,R0 & BOZ $+3   ;IF SEEK COMP. THIS HAIO ON...
00C36 40130 33C3A  6205              AND AQ,FOUTL,R3,R0 & BNZ $+4   ;DRIVES 0-23, JUMP
00C37 FFFFF 33C3A  6206              BNZ $+3                        ;DRIVES 24-31, JUMP
00C38 0B3F6 FFFFF  6207              INCRB R6                       ;IF NOT, INCR. HAIO NO...
00C39 337F2 E0C2C  6208              OR ZB,LSBZ,,R2 & JMP POLL21    ;AND TRY NEXT HAIO
                6209    ;
00C3A 3705F F0C3C  6210              RDMSKQ $+2                     ;COMPUTE HAIO NO...
00C3B 0016E FFFFF  6211              ADD AQ,FOUTL,R6,IREG           ;TO SELECT
00C3C FFFFF 6F7FF  6212              JIR
                6213    ;
00C3D E407E E0C41  6214              CTL HIO,SPILCH1 & JMP $+4      ;SELECT HAIO-1
00C3E E407D E0C41  6215              CTL HIO,SPILCH2 & JMP $+3      ;SELECT HAIO-2
00C3F E407B E0C41  6216              CTL HIO,SPILCH3 & JMP $+2      ;SELECT HAIO-3
00C40 E4077 FFFFF  6217              CTL HIO,SPILCH4                ;SELECT HAIO-4
00C41 3412F AF02D  6218              ALDREG R2,RAME & RTS EFALLOW   ;NEXT E.F. MUST BE ON THIS HAIO
                6219    ;
                6220    ;    IT WAS IMPOSSIBLE TO DETERMINE WHICH SPI TO SELECT.  THE
                6221    ;PENDING INTERRUPT WILL BE CLEARED AND NOT SENT AT ALL.
                6222    ;
00C42 37352 F0054  6223    POLL23:   RDMSKB R2,SKPEND               ;LOC. OF SEEK COMP. PENDING
00C43 FFFFF 1F1DC  6224              JSR DELABIT                    ;CLEAR THE BIT FOR THIS DRIVE
00C44 FFFFF E000F  6225              JMP WAIT05                     ;RETURN TO THE WAIT LOOP
                6226
                6227
                6228    ;COMPUTE THE DRIVE NUMBER - POLLING SEQUENCE.
                6229    ;UPON ENTRY THE BIT FOR THIS DRIVE MUST BE IN R0 (0-23) OR Q (24-31).
                6230    ;UPON EXIT:  R3  = BIT FOR THIS DRIVE (24-31).
                6231    ;            R4  = BIT FOR THIS DRIVE ( 0-23).
                6232    ;            R13 = LOGICAL DRIVE NUMBER
                6233    ;Return to Address Reg. (AR) when complete.
                6234
00C45 433FD FFFFF  6235    DRVNBR:   RESETR R13                     ;DELETE CURRENT DRIVE NO.
00C46 433F3 FFFFF  6236              RESETR R3                      ;CLEAR MSB'S
00C47 43FF4 FFFFF  6237              AND ZB,LSB1,,R4                ;1ST DRIVE TO CHECK IS 0
00C48 41140 FFFFF  6238    DRVNBR1:  AND AB,FOUTL,R4,R0             ;CHECK THIS DRIVE (0-23)
00C49 333F4 7344A  6239              OR ZB,B,,R4 & CJARB ALUZ,$+1   ;IF DRIVE 0-23, RETURN
00C4A 337F4 3344C  6240              OR ZB,LSBZ,,R4 & BOZ DRVNBR2   ;IF DRIVE NO.> 23, JUMP
00C4B 0B3FD E0C48  6241              INCRB R13 & JMP DRVNBR1        ;INCR. DRIVE NO. & CHECK NEXT DRIVE
                6242    ;
00C4C 43FF3 FFFFF  6243    DRVNBR2:  AND ZB,LSB1,,R3                ;NEXT DRIVE TO CHECK IS 24
00C4D 40130 FFFFF  6244              AND AQ,FOUTL,R3,R0             ;CHECK THIS DRIVE (24-31)
00C4E FFFFF 7344F  6245              CJARB ALUZ,$+1                 ;IF DRIVE 24-31, RETURN
00C4F 0B3FD FFFFF  6246              INCRB R13                      ;INCR. DRIVE NO.
00C50 337F3 E0C4D  6247              OR ZB,LSBZ,,R3 & JMP DRVNBR2+1 ;SHIFT BIT TO CHECK NEXT DRIVE
                6248
                6249
                6250    TITLE  ASSEMBLE AND SEND SENSE BYTES
```

Addr Line ASSEMBLE AND SEND SENSE BYTES

```
                    6252
                    6253
                    6254    ;     THE SENSE BYTE ROUTINE WILL ASSEMBLE THE SENSE BYTES SAVED IN SMB
                    6255    ;MEMORY INTO A-FORMAT (EVERY 9TH BIT = 0) WORDS TO BE SENT TO THE CHANNEL.
                    6256    ;     BYTES 0-7 ARE ALWAYS ASSEMBLED FROM THE SAME LOCATIONS IN SMB MEMORY.
                    6257    ;BYTES 8-23 WILL BE ASSEMBLED ACCORDING TO THE FORMAT DECODE OF BYTE 7.
                    6258    ;
                    6259    ;     WHEN 4 BYTES HAVE BEEN ASSEMBLED, ONE WORD WILL BE SENT TO THE
                    6260    ;CHANNEL.  THIS WILL REPEAT UNTIL 6 WORDS (24 BYTES) HAVE BEEN SENT.
                    6261    ;
                    6262    ;     THE REGISTERS ARE:
                    6263    ;       R0 & Q = SENSE WORD ASSEMBLY REGISTERS
                    6264    ;       R1,R2 = SCRATCH
                    6265    ;       R3 = BYTE POINTER (ADDRESS OF SENSE BYTE IN SMB)
                    6266    ;       R4 = ASSEMBLY COUNT (# OF CONSEQUTIVE BYTES TO ASSEMBLE)
                    6267    ;       R5 = NOT USED
                    6268    ;       R6 = BYTE COUNT (# OF BYTES TO ASSEMBLE ONE WORD)
                    6269    ;       R7 = SENSE BYTE 7
                    6270    ;       R8 = DISPLAY ADDRESS REG.
                    6271    ;       R9 = SCRATCH
                    6272    ;       R10 = STATUS ACTION CODE
                    6273    ;       R11 = NOT USED
                    6274    ;       R12 = CURRENT COMMAND CODE
                    6275    ;       R13 = DRIVE NO.
                    6276    ;       R14,R15 = FLAG REGISTERS
                    6277
                    6278
                    6279    SENS00:   JSR.P DOUBLE                    ;DOUBLE DRIVE NO. IF NEEDED
00C51 E0403 1F125   6279 +            SETPGE DOUBLE_SHR_H#8 & JSRP DOUBLE
00C52 433F0 FFFFF   6280              RESETR R0                       ;RESET SENSE ASSEMBLY REGISTER
00C53 433F3 FFFFF   6281              RESETR R3                       ;SET SENSE BYTE = 0
00C54 37356 F0004   6282              RDMSKB R6,4                     ;BYTES PER WORD = 4
00C55 37358 F02B8   6283              RDMSKB R8,2B8                   ;DISPLAY SENSE AT 4E8
00C56 37354 F0008   6284              RDMSKB R4,8                     ;ASSEMBLY COUNT = 8
00C57 FFFFF 1F508   6285              JSR SENS50                      ;ASSEMBLE & SEND 8 BYTES
                    6286    ;
00C58 373F7 F0007   6287              RDRAMB R7,SEN07                 ;GET SENSE BYTE 7
00C59 55357 F000F   6288              RSTRBIT R7,H#F                  ;DELETE MESSAGE NO.
00C5A FFFFF 33C5F   6289              BNZ SENS02                      ;IF NOT MESSAGE 0, JUMP
00C5B 37353 F0FFF   6290    SENS01:   RDMSKB R3,H#FFF                 ;NEXT BYTES ARE ZEROS
00C5C 37354 F0010   6291              RDMSKB R4,D#16                  ;ASSEMBLY COUNT = 16
00C5D FFFFF 1F508   6292              JSR SENS50                      ;ASSEMBLE & SEND 16 ZERO BYTES
00C5E FFFFF E0CF4   6293              JMP SENS40                      ;SEND ENDING STATUS
                    6294    ;
00C5F 65057 F0010   6295    SENS02:   TSTRMSK R7,10                   ;SENSE FORMAT 1?
00C60 FFFFF 33C6B   6296              BNZ SENS05                      ;IF NOT, CHECK FOR FORMAT 2
                    6297    ;
                    6298    ;   Sense Format 1 (Disk Equip. Check) assembly
                    6299    ;
00C61 37353 F0008   6300              RDMSKB R3,SEN108                ;SENSE 8, FORMAT 1 IS NEXT
00C62 37354 F0008   6301              RDMSKB R4,8                     ;ASSEMBLY COUNT = 8
00C63 FFFFF 1F508   6302              JSR SENS50                      ;ASSEMBLE & SEND 8 BYTES
                    6303    ;
```

Addr Line ASSEMBLE AND SEND SENSE BYTES

```
00C64 37354 F0005  6304            RDMSKB R4,5                      ;ASSEMBLY COUNT = 5
00C65 37353 F0FFF  6305   SENS03:   RDMSKB R3,H#FFF                  ;NEXT BYTES ARE ZEROS
00C66 FFFFF 1F508  6306            JSR SENS50                       ;ASSEMBLE AND SEND ZEROS
00C67 37353 F0010  6307            RDMSKB R3,SEN121                 ;FORMAT 1,2 - BYTE 21 IS NEXT
00C68 0F304 1F508  6308            ADD DZC,B,ZERO,R4 & JSR SENS50   ;ASSEMBLE IT
00C69 37354 F0002  6309            RDMSKB R4,2                      ;NEXT 2 BYTES ARE ZEROS
00C6A FFFFF E0C8C  6310            JMP SENS08                       ;ASSEMBLE & SEND THEM
                   6311   ;
00C6B 65057 F0020  6312   SENS05:   TSTRMSK R7,20                    ;SENSE FORMAT 2?
00C6C FFFFF 33C7E  6313            BNZ SENS07                       ;IF NOT, CHECK FOR 4 OR 5
                   6314   ;
                   6315   ;   Sense Format 2 (C.U. Equip. Check) assembly
                   6316   ;
00C6D 37353 F0020  6317            RDMSKB R3,SEN208                 ;FORMAT 2, BYTE 8 IS NEXT
00C6E 37354 F0005  6318            RDMSKB R4,5                      ;ASSEMBLY COUNT IS 5
00C6F FFFFF 1F508  6319            JSR SENS50                       ;ASSEMBLE & SEND 5 MORE BYTES
00C70 37353 F000D  6320            RDMSKB R3,SEN113                 ;FORMAT 1,2 - BYTE 13 IS NEXT
00C71 37354 F0003  6321            RDMSKB R4,3                      ;ASSEMBLY COUNT IS 3
00C72 FFFFF 1F508  6322            JSR SENS50                       ;ASSEMBLE 3 MORE
00C73 37353 F0025  6323            RDMSKB R3,SEN216                 ;NEXT IS FORMAT 2, BYTE 16
00C74 0F304 1F508  6324            ADD DZC,B,ZERO,R4 & JSR SENS50   ;ASSEMBLE IT
00C75 37353 F0500  6325            RDMSKB R3,SEN217                 ;Sense Format 2, Byte 17
00C76 37354 F0004  6326            RDMSKB R4,4                      ;Asembly count = 4
00C77 FFFFF 1F508  6327            JSR SENS50                       ;Assemble next 4 bytes
00C78 37353 F0010  6328            RDMSKB R3,SEN121                 ;Next byte is Format 1,2 - Byte 21
00C79 0F304 1F508  6329            ADD DZC,B,ZERO,R4 & JSR SENS50   ;Assemble it
00C7A 37353 F0504  6330            RDMSKB R3,SEN222                 ;Sense Format 2, Byte 22
00C7B 37354 F0002  6331            RDMSKB R4,2                      ;Assembly count = 2
00C7C FFFFF 1F508  6332            JSR SENS50                       ;Assemble last 2 bytes and...
00C7D FFFFF E0CF4  6333            JMP SENS40                       ;Send Ending status
                   6334   ;
00C7E 65057 F0040  6335   SENS07:   TSTRMSK R7,40                    ;SENSE FORMAT 4?
00C7F FFFFF 33484  6336            BOZ     SENS07A                  ;If yes, jump
00C80 65057 F0050  6337            TSTRMSK R7,50                    ;SENSE FORMAT 5?
00C81 FFFFF 33CA0  6338            BNZ SENS10                       ;IF NOT, CHECK FOR 6
00C82 45057 F0008  6339            TSTRBIT  R7,8                    ;Cache. ECC Corr. (58 or 59)?
00C83 FFFFF 33C58  6340            BNZ     SENS01                   ;Yes, rest of the bytes are zeros
                   6341   ;
                   6342   ;   Sense Format 4 or 5 assembly.
                   6343   ;
00C84 37353 F0011  6344   SENS07A:  RDMSKB R3,SEN408                 ;NEXT IS FORMAT 4,5 - BYTE 8
00C85 37354 F0007  6345            RDMSKB R4,7                      ;ASSEMBLY COUNT = 7
00C86 FFFFF 1F508  6346            JSR SENS50                       ;ASSEMBLE & SEND 7 BYTES
                   6347   ;
00C87 65157 F0040  6348            TSTREQU R7,H#40                  ;SENSE FORMAT 4?
00C88 FFFFF 33C8F  6349            BNZ SENS09                       ;IF NOT, MUST BE 5
                   6350   ;
                   6351   ;   Sense Format 4 (ECC Uncorrctable) assembly
                   6352   ;
00C89 37353 F0018  6353            RDMSKB R3,SEN415                 ;NEXT IS SENSE 15, FORMAT 4
00C8A 0F704 1F508  6354            ADD DZC,LSBZ,ZERO,R4 & JSR SENS50 ;ASSEMBLE 2 MORE BYTES
00C8B 37354 F0007  6355            RDMSKB R4,7                      ;ASSEMBLY COUNT = 7
00C8C 37353 F0FFF  6356   SENS08:   RDMSKB R3,H#FFF                  ;NEXT BYTES ARE ZEROS
00C8D FFFFF 1F508  6357            JSR SENS50                       ;ASSEMBLE & SEND ZEROS BYTES
00C8E FFFFF E0CF4  6358            JMP SENS40                       ;SEND ENDING STATUS
```

```
Addr            Line  ASSEMBLE AND SEND SENSE BYTES

                6359  ;
                6360  ;   SENSE FORMAT 5 (ECC Correctable) ASSEMBLY
                6361  ;
00C8F 34302 FFFFF  6362  SENS09:  RTOR R0,R2                    ;TEMP. SAVE SENSE ASSY. REG. IN R2
00C90 323F3 FFFFF  6363          QTOR R3                       ;SENSE ASSY. MSB'S TO R3
00C91 373F0 F02CD  6364          RDRAMB R0,RECLNTH             ;GET THE RECORD LENGTH AND...
                6365             JSR.P FOUR5DN                 ;CONVERT IT TO BYTES
00C92 E0400 1F368  6365 +        SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
00C93 45050 F00FF  6366          TSTRBIT R0,H#FF               ;ISOLATE LSB'S IN Q
00C94 321FF F001C  6367          QLDRAM SEN517                 ;SAVE THEM FOR DISPLAY
00C95 3491F F001B  6368          ALDRAMU R1,SEN516             ;SAVE MSB'S FOR DISPLAY
                6369  ;
00C96 3403F FFFFF  6370          RTOQ R3                       ;RESTORE ASSEMBLY MSB'S
00C97 37353 F001A  6371          RDMSKB R3,SEN515              ;NEXT BYTE IS FORMAT 5, BYTE 15
00C98 37354 F0003  6372          RDMSKB R4,3                   ;ASSEMBLY COUNT = 3
00C99 34320 1F508  6373          RTOR R2,R0 & JSR SENS50       ;RESTORE ASSY. LSB'S & ASSEMBLE 3
00C9A 37353 F0FFF  6374          RDMSKB R3,H#FFF               ;NEXT BYTES ARE ZEROS
00C9B 37354 F0003  6375          RDMSKB R4,3                   ;THERE ARE 3 OF THEM
00C9C FFFFF 1F508  6376          JSR SENS50                    ;ASSEMBLE THEM
00C9D 37353 F001D  6377          RDMSKB R3,SEN521              ;NEXT BYTE IS FORMAT 5, BYTE 21
00C9E 0F704 1F508  6378          ADD DZC,LSBZ,ZERO,R4 & JSR SENS50 ;ASSEMBLE 2 BYTES
00C9F 0F304 E0C8C  6379          ADD DZC,8,ZERO,R4 & JMP SENS08 ;LAST BYTE IS ZEROS
                6380  ;
00CA0 65057 F0060  6381  SENS10:  TSTRMSK R7,60                 ;SENSE FORMAT 6?
00CA1 FFFFF 33CA9  6382          BNZ SENS12                    ;IF NOT, CHECK FOR 8
                6383  ;
                6384  ;   Sense Format 6 (Error/usage log) assembly
                6385  ;
00CA2 FFFFF 1F516  6386          JSR SENS70                    ;ASSEMBLE ERROR/USAGE BYTES
00CA3 37354 F000C  6387          RDMSKB R4,D#12                ;ASSEMBLY COUNT = 12
00CA4 433F0 1F508  6388          RESETR R0 & JSR SENS50        ;ASSEMBLE & SEND 12 BYTES
                6389  ;
00CA5 3419E C0008  6390          ALDREG R9,IREG & LDCT 8       ;1ST LOG LOC. TO CLEAR
                6391             JSR.P CLR25                   ;CLEAR ERROR/USAGE LOG THIS DRIVE
00CA6 E0400 1F33D  6391 +        SETPGE CLR25_SHR_H#B & JSRP CLR25
00CA7 37354 F0004  6392          RDMSKB R4,4                   ;ASSEMBLY COUNT = 4
00CA8 FFFFF E0C8C  6393          JMP SENS08                    ;SEND 4 BYTES OF ZEROS
                6394  ;
00CA9 65057 F0080  6395  SENS12:  TSTRMSK R7,80                 ;CHECK FOR FORMAT 8
00CAA FFFFF 33C5B  6396          BNZ SENS01                    ;IF IT'S NOT, MUST BE FORMAT 9
00CAB 37353 F0012  6397          RDMSKB R3,SEN409              ;LOC. OF FIRST FORMAT 8 BYTE
00CAC 37354 F0010  6398          RDMSKB R4,D#16                ;ASSEMBLY COUNT = 16
00CAD FFFFF 1F508  6399          JSR SENS50                    ;ASSEMBLE & SEND 16 BYTES
00CAE FFFFF E0CF4  6400          JMP SENS40                    ;SEND ENDING STATUS
                6401
                6402  ;   READ AND RESET BUFFERED LOG
                6403  ;
00CAF 37357 F0060  6404  SENS15:  RDMSKB R7,60                 ;SENSE 7 = ERROR/USAGE COUNT
                6405             SETMBIT D#11,SEN03            ;Sense 3 = Recov. Action 11
00CB0 37350 F000B  6405 +        RDMSKB   R0,D#11
00CB1 3410F F0003  6405 +        ALDRAM   R0,SEN03
00CB2 3417F F0007  6406          ALDRAM R7,SEN07               ;SAVE SENSE 7
```

Addr Line ASSEMBLE AND SEND SENSE BYTES

```
                    6407            SETMBIT 10,SEN02                ;Sense 2 = Environmental Data
00CB3 37350 F0010   6407 +          RDMSKB   R0,10
00CB4 3410F F0002   6407 +          ALDRAM   R0,SEN02
00CB5 FFFFF E0C51   6408            JMP SENS00
                    6409
                    6410    ;   READ CONFIGURATION
                    6411    ;
00CB6 0F10F F03E5   6420    SENS18: ADDC DZ,FOUTL,ZERO,RAME & NOP LASTFLG ;PRESET LAST DRIVE FLAG
00CB7 343D7 1F169   6421            RTOR R13,R7 & JSR SEL00$        ;SELECT THE DRIVE
00CB8 3715E F0012   6423            MOVE MSK2,1REG & NOP SEN409
00CB9 373E3 C000F   6424            RDREGB IREG,R3 & LDCT 15        ;START OFF BY CLEARING ...
                    6425            JSR.P CLR25                     ;THE CONFIGURATION BYTES
00CBA E0400 1F33D   6425 +          SETPGE CLR25_SHR_H#B & JSRP CLR25
                    6426    ;
00CBB 37350 F0050   6427            RDMSKB R0,50
00CBC 3437D 1F4D9   6428            RTOR R7,R13 & JSR SENS25        ;LOGICAL DRIVE BACK TO R13
00CBD 37350 F0056   6429            RDMSKB R0,56                    ;CONTROL UNIT TYPE = 5056
00CBE FFFFF 1F4D9   6430            JSR SENS25
00CBF 37353 F0018   6431            RDMSKB R3,SEN415                ;NEXT CONFIG. BYTE IS 14
00CC0 37350 F0084   6432            RDMSKB R0,84                    ;DRIVE IS 84__
00CC1 FFFFF 1F4D9   6433            JSR SENS25
                    6434    ;
00CC2 FFFFF E1B47   6436            JMP     SENS80                  ;$
                    6444    ;
00CC3 37350 F0070   6446    SENS18A: RDMSKB R0,70                   ;<EMULATION IS 8470
                    6459    ;
00CC4 37353 F0019   6460    SENS19: RDMSKB R3,SEN416
00CC5 FFFFF 1F4D9   6461            JSR SENS25                      ;SAVE EMULATION
                    6466    ;
00CC6 45058 F000E   6467            TSTRBIT R8,H#E                  ;ISOLATE DRIVE TYPE IN Q
00CC7 325F0 1F4D9   6468            OR ZQ,RSBZ,,R0 & JSR SENS25     ;SAVE DRIVE TYPE
00CC8 60580 4F002   6469            XOR AQ,RSBZ,R8,R0 & PSHLD 2
00CC9 335F0 887FF   6470            RSHFTZ R0                       ;MOVE RECORD LENGTH INTO POSITION
00CCA 45350 F0003   6471            MSKRBIT  R0,3                   ;ISOLATE LENGTH FLAGS
00CCB 35350 F0008   6473            SETRBIT R0,8                    ;<INCLUDE NATIVE MODE FLAG
                    6482    ;
00CCC FFFFF 1F4D9   6490    SENS20: JSR SENS25                      ;SAVE LENGTH FLAGS
```

ASSEMBLE AND SEND SENSE BYTES

```
                    6492    ;
00CCD 37353 F001E   6493            RDMSKB R3,SEN522                ;LOC. OF MONTH
00CCE 37350 F0000   6494            RDMSKB R0,MONTH
00CCF FFFFF 1F4D9   6495            JSR SENS25                      ;SAVE THE MONTH
00CD0 37350 F0016   6496            RDMSKB R0,DAY
00CD1 FFFFF 1F4D9   6497            JSR SENS25                      ;DAY AND...
00CD2 37350 F0090   6498            RDMSKB R0,YEAR
00CD3 FFFFF 1F4D9   6499            JSR SENS25                      ;YEAR OF THIS ASSEMBLY
00CD4 37350 F000B   6500            RDMSKB R0,REVISION
00CD5 FFFFF 1F4D9   6501            JSR SENS25                      ;SAVE REVISION LEVEL
00CD6 3705F F0080   6502            RDMSKQ 80                       ;SENSE 7 IS...
00CD7 321FF F0007   6503            QLDRAM SEN07                    ;FORMAT 8, MESSAGE 0
00CD8 FFFFF E0C51   6504            JMP SENS00                      ;ASSEMBLE & SEND CONFIGURATION
                    6505    ;
00CD9 3413E FFFFF   6506    SENS25: ALDREG R3,IREG                  ;LOAD LOC. OF BYTE
00CDA 3410D FFFFF   6507            ALDREG R0,RAMI                  ;SAVE THE DATA
00CDB 0B3F3 AF7FF   6508            INCRB R3 & RTS                  ;INCR. LOC. & RETURN
                    6509
                    6510
                    6511    ;   DISPLAY SENSE DATA (4E8-4ED) & SEND ONE WORD TO THE CHANNEL
                    6512    ;
00CDC 3418E FFFFF   6513    SENS30: ALDREG R8,IREG                  ;LOAD DISPLAY ADDR.
00CDD 321FD 1F4F2   6514            QLDREG RAMI & JSR SENS33        ;DISPLAY 12 MSB'S
00CDE 3490D 1F4F2   6515            ALDREGU R0,RAMI & JSR SENS33    ;DISPLAY 12 MIDDLE BITS
00CDF 3410D 1F4F2   6516            ALDREG R0,RAMI & JSR SENS33     ;DISPLAY 12 LSB'S
                    6517    ;
00CE0 721FC FFFFF   6518            NQLDREG UBR3                    ;LOAD BITS 24-35 TO U-BUS
00CE1 7490B FFFFF   6519            NALDREGU R0,UBR2                ;LOAD BITS 12-23 TO U-BUS
00CE2 7410A FFFFF   6520            NALDREG R0,UBR1                 ;LOAD BITS  0-11 TO U-BUS
00CE3 E4401 FFFFF   6521            CTL HIO,HDRVGTS                 ;ENABLE THE WORD TO THE U-BUS
00CE4 E4404 FFFFF   6522            CTL HIO,HIOTGTS                 ;ENABLE IDR TAG
00CE5 E4202 FFFFF   6523            CTL HIO,SETIDRS                 ;SET IDR
00CE6 37352 F1004   6524            RDMSKB R2,1004                  ;ALLOW I.A. & E.F.
00CE7 E4200 FFFFF   6525            CTL HIO,RSTIDR
00CE8 37356 F1388   6526            RDMSKB R6,1388                  ;SET 2MS. TIMER
                    6527    ;
00CE9 45082 334EF   6528            AND DB,Q,HIOST,R2 & BOZ SENS31  ;CHECK EF/IA; IF TIME-OUT, JUMP
00CEA 133F6 334E9   6529            DECRB R6 & BOZ $-1              ;DECR. TIMER; IF NO EF/IA, BACK
                    6530    ;
00CEB 4605F F0004   6531            TSTQBIT 4                       ;CHECK FOR E.F.
00CEC E0204 334F0   6532            CTL SMC,RSTIDAK & BOZ SENS32    ;IF NOT, CONTINUE; RESET I.A.
00CED 3535E F0002   6533            SETRBIT R14,2                   ;SET FORCED E.F. FLAG
                    6534            JSR.P CLREF                     ;CLEAR THE E.F.
00CEE E0403 1F583   6534 +          SETPGE CLREF_SHR_H#B & JSRP CLREF
                    6535    ;
00CEF 3535F F0002   6536    SENS31: SETRBIT R15,2                   ;SET TRUNCATION FLAG
                    6537    ;
00CF0 E4402 FFFFF   6538    SENS32: CTL HIO,HDRVGTR                 ;RESET DATA ON U-BUS
00CF1 37346 AF004   6539            RDREGB MSK1,R6 & RTS D#4        ;BYTES PER WORD = 4 & RETURN
                    6540    ;
00CF2 0B3F8 FFFFF   6541    SENS33: INCRB R8                        ;INCR. DISPLAY ADDRESS
00CF3 3418E AF7FF   6542            ALDREG R8,IREG & RTS            ;LOAD IT & RETURN
                    6543
```

```
 Addr            Line  ASSEMBLE AND SEND SENSE BYTES

                 6544
                 6545   ;    SEND ENDING STATUS
                 6546   ;
00CF4 E4408 FFFFF 6547   SENS40:   CTL HIO,HIOTGTR                ;DISABLE IDR TAG
00CF5 3735A F0008 6548            RDMSKB R10,8                   ;STATUS ACTION = 8
00CF6 37360 FFFFF 6549            RDREGB SMCST,R0                ;READ XMC STATUS
00CF7 45340 F0008 6550            AND DB,B,MSK1,R0 & NOP H#8     ;ISOLATE SMB PARITY IN R0
00CF8 6505C F0004 6551            TSTRMSK R12,4                  ;SENSE I/O COMMAND?
00CF9 333F0 33D03 6552            OR ZB,B,,R0 & BNZ SENS41       ;NO, JUMP; CHECK SMB PARITY
00CFA FFFFF 334FD 6553            BOZ $+3                        ;IF NO ERROR, CONTINUE
00CFB 3535D F0200 6554            SETRBIT R13,200                ;INCLUDE UNIT CHECK WITH STATUS
00CFC 0B3FA E0D36 6555   SENS40A:  INCRB R10 & JMP END00          ;STATUS ACTION = 9 & SEND STATUS
                 6556   ;
00CFD 4505F F0002 6557            TSTRBIT R15,2                  ;TRUNCATION FLAG?
00CFE FFFFF 33CFC 6558            BNZ SENS40A                    ;YES, JUMP
00CFF 4505D F007F 6559            TSTRBIT R13,7F                 ;ISOLATE THE DRIVE NO. AND...
00D00 3635D F0C00 6560            OR DQ,B,MSK2,R13 & NOP H#C00   ;INCLUDE CHAN. END/DEV END
00D01 37359 FFFFF 6561            RDMSKB R9,H#FFFF               ;SET LARGE POLLING DELAY TIMER
00D02 FFFFF E0D3A 6562            JMP END01+2                    ;SEND NORMAL STATUS
                 6563   ;
                 6564   SENS41:   BNZ.P READ62                   ;IF SMB PARITY, EQUIP. CHECK
00D03 E0400 33CAD 6564 +          SETPGE READ62_SHR_H#B & BNZP READ62
00D04 3735A F0007 6565            RDMSKB R10,7                   ;STATUS ACTION = 7
00D05 4505F F0002 6566            TSTRBIT R15,2                  ;CHANNEL TRUNCATION?
                 6567             BOZ.P NCMPLT1                  ;IF NOT, SEND NORMAL STATUS
00D06 E0400 33093 6567 +          SETPGE NCMPLT1_SHR_H#B & BOZP NCMPLT1
00D07 FFFFF E04A7 6568            JMP READ24                     ;ELSE, SEND TRUNCATION STATUS
                 6569
                 6570
                 6571   ;    ASSEMBLE SENSE BYTES INTO A-FORMAT WORDS.
                 6572   ;
00D08 65153 F0FFF 6573   SENS50:   TSTREQU R3,H#FFF               ;IS THIS BYTE ZERO?
00D09 3413E 3350C 6574            ALDREG R3,IREG & BOZ $+3       ;IF YES, JUMP; LOAD BYTE ADDR.
00D0A 0B3F3 FFFFF 6575            INCRB R3                       ;INCR. BYTE ADDRESS
00D0B 353D0 E0D0D 6576            OR DB,B,RAMI,R0 & JMP $+2      ;MERGE THIS BYTE WITH OTHERS
00D0C 55350 F01FF 6577            RSTRBIT R0,1FF                 ;ZEROS TO THIS BYTE
00D0D 55350 F0100 6578            RSTRBIT R0,100                 ;RESET BIT 8
00D0E 133F6 FFFFF 6579            DECRB R6                       ;DECR. BYTES PER WORD
00D0F 343F1 134DC 6580            RTOR R15,R1 & CJSR ALUZ,SENS30 ;IF 0, SEND A WORD; FLAGS TO R1
                 6581   ;
00D10 336F0 4F007 6582            OR ZB,LSBQZ,,R0 & PSHLD 7      ;LEFT SHIFT...
00D11 336F0 887FF 6583            LSHFTDZ R0                     ;THE DATA 9 PLACES
00D12 45151 F0002 6584            TSTBIT R1,2                    ;TRUNCATION FLAG?
00D13 133F4 B3CF4 6585            DECRB R4 & CJPOP NALUZ,SENS40  ;IF SET, JUMP; DECR. ASSY. COUNT
00D14 FFFFF 33D08 6586            BNZ SENS50                     ;IF NOT 0, GET NEXT BYTE
00D15 FFFFF AF7FF 6587            RTS                            ;ELSE, RETURN
```

Addr Line ASSEMBLE AND SEND SENSE BYTES

```
                    6589
                    6590    ;    CONVERT ERROR/USAGE LOG TO BYTES FOR SENSE BYTE ASSEMBLY
                    6591    ;
                    6592    SENS70:  JSR.P ERRLOG                       ;LOC. OF ERROR/USAGE LOG
00D16 E0400 1F350   6592 +           SETPGE ERRLOG_SHR_H#B & JSRP ERRLOG
00D17 34329 FFFFF   6593             RTOR R2,R9                         ;SAVE LOC. TO CLEAR LOG
00D18 37354 F0003   6594             RDMSKB R4,3                        ;PRESET ITERATION COUNT = 3
00D19 37353 F01E0   6595             RDMSKB R3,1E0                      ;SET LOC. OF 1ST BYTE
                    6596    ;
00D1A 3412E 1F0CD   6597    SENS71:  ALDREG R2,IREG & JSR SMBRDA        ;GET THE NEXT LOG WORD
00D1B FFFFF 1F52C   6598             JSR SENS75                         ;LOAD BITS 24-31
00D1C 05343 F0003   6599             ADD DB,B,MSK1,R3 & NOP D#3         ;INCR. BYTE LOC.+ 3
00D1D 3400F 1F52C   6600             RTOQ R0 & JSR SENS75               ;LOAD BITS 0-7
00D1E 1D343 F0002   6601             SUBA DB,B,MSK1,R3 & NOP D#2        ;DECR. BYTE LOC.- 2
00D1F FFFFF 4F003   6602             PSHLD 3
00D20 335F0 887FF   6603             RSHFTZ R0                          ;MOVE BITS 16-23 INTO POSITION
00D21 3490E C0003   6604             ALDREGU R0,IREG & LDCT 3
00D22 370EF 1F52C   6605             RDREGQ IREG & JSR SENS75           ;LOAD BITS 16-23
00D23 335F0 98523   6606             OR ZB,RSBZ,,R0 & CJPCNT EVCNTZ,$   ;MOVE BITS 8-15 INTO POSITION
00D24 0B3F3 FFFFF   6607             INCRB R3                           ;INCR. BYTE LOC.+ 1
00D25 3400F 1F52C   6608             RTOQ R0 & JSR SENS75               ;LOAD BITS 8-15
                    6609    ;
00D26 133F4 FFFFF   6610             DECRB R4                           ;DECR. ITERATION COUNT
00D27 FFFFF 3352B   6611             BUZ SENS73                         ;IF COUNT = 0, JUMP
00D28 05342 F0003   6612             ADD DB,B,MSK1,R2 & NOP D#3         ;LOC. OF NEXT LOG WORD
00D29 05343 F0002   6613             ADD DB,B,MSK1,R3 & NOP D#2         ;INCR. BYTE LOC.+ 2
00D2A FFFFF E0D1A   6614             JMP SENS71                         ;ASSEMBLE NEXT 4 BYTES
                    6615    ;
00D2B 37343 AF1E0   6616    SENS73:  RDREGB MSK1,R3 & RTS H#1E0         ;PRESET BYTE POINTER & RETURN
                    6617    ;
00D2C 3413E FFFFF   6618    SENS75:  ALDREG R3,IREG                     ;LOAD THE ADDRESS
00D2D 4614D AF0FF   6619             AND DQ,FOUTL,MSK1,RAMI & RTS H#FF  ;MASK & WRITE ONE BYTE; RETURN
                    6620
                    6621
                    6622    ;    This subroutine clears all sense bytes except 4.  Only IREG is used.
                    6623    ;
00D2E 3710E C0003   6624    CLRSENS: MOVE ZERO,IREG & LDCT 3            ;Starting at Sense Byte 0...
                    6625             JSR.P CLR25                        ;Clear first 4 bytes
00D2F E0400 1F33D   6625 +           SETPGE CLR25_SHR_H#B & JSRP CLR25
00D30 3715E F0005   6626             MOVE MSK2,IREG & NOP SEN05         ;Next byte is 5
00D31 FFFFF C0020   6627             LDCT 32
                    6628             JSR.P CLR25                        ;Clear the rest of the Sense
00D32 E0400 1F33D   6628 +           SETPGE CLR25_SHR_H#B & JSRP CLR25
00D33 3715E F0500   6629             MOVE MSK2,IREG & NOP SEN217        ;Loc. of new Sense bytes
00D34 FFFFF C0005   6630             LDCT 5                             ;There are 6 of them
00D35 FFFFF E033D   6631             JMP CLR25                          ;Clear them & return
                    6632    ;
                    6633
                    6635    TITLE CDC O.S. - ENDING ROUTINE
```

Addr Line CDC O.S. - ENDING ROUTINE

Addr Line CDC O.S. - ENDING ROUTINE

```
                    6644
                    6645    ;UPON ENTRY:  R3 = RECOVERY ACTION
                    6646    ;             R10= STATUS ACTION CODE  (R9 IF UNIT CHECK)
                    6647    ;             R13= DRIVE NUMBER
                    6648    ;REGISTERS USED:  R0,R1 & R2 = SCRATCH
                    6649    ;         R3 = RECOVERY ACTION OR...
                    6650    ;              BIT FOR THIS DRIVE (24-31)
                    6651    ;         R4 = BIT FOR THIS DRIVE (0-23)
                    6652    ;         R5,R6 = USED IF UNIT CHECK,
                    6653    ;                 NOT USED WITH OTHER STATUSES
                    6654    ;         R7 = ILD FLAGS
                    6655    ;         R8 = BIT FOR THIS DRIVE (24-31)
                    6656    ;         R9 = POLLING DELAY TIMER FOR WAIT LOOP
                    6657    ;         R10= STATUS ACTION CODE
                    6658    ;         R11= 3 SECOND E.I. TIME-OUT TIMER
                    6659    ;         R12= HEAD NO. IF UNIT CHECK
                    6660    ;            NOT USED WITH OTHER STATUSES
                    6661    ;         R13= STATUS (24 LSB'S)
                    6662    ;         R14,R15 = FLAGS
                    6663
                    6664
                    6665    ;
                    6666    ;     STATUS:   CHANNEL END/DEVICE END
                    6667    ;
00D36 4535D F00FF   6668    END00:    NSKRBIT  R13,H#FF                ;DELETE ANY EXISTING STATUS
00D37 3535D F0400   6669              SETRBIT R13,400                  ;STATUS IS DEVICE END
00D38 3535D F0800   6670    END01:    SETRBIT R13,800                  ;STATUS IS CHANNEL END
00D39 373F9 F03E4   6671              RDRAMB R9,DLAYDE                 ;1 MSEC. DEV. END DELAY TIMER
00D3A 373F7 F00BA   6672              RDRAMB R7,ILDFLG                 ;GET ILD FLAGS
00D3B 45057 F0004   6673              TSTRBIT R7,4                     ;CHECK ILD RUNNING
00D3C FFFFF 33D79   6674              BNZ END50                        ;IF IT IS, JUMP
                    6675    ;
00D3D 4535F F0002   6679              NSKRBIT R15,2                    ;CLEAR ALL FLAGS EXCEPT TRUNCATION
00D3E FFFFF E0D79   6680              JMP END50
                    6681    ;
                    6682    ;     STATUS:   UNIT CHECK
                    6683    ;
00D3F 37359 F000B   6684    END10:    RDMSKB R9,0B                     ;STATUS ACTION = 0B
00D40 3535D F0200   6685    END10A:   SETRBIT R13,200                  ;ADD UNIT CHECK STATUS
00D41 3419F F025F   6686              ALDRAM R9,STATACT                ;SAVE STATUS ACTION FOR LATER
00D42 3413F F0003   6687              ALDRAM R3,SEN03                  ;SAVE RECOVERY ACTION FOR SENSE
                    6695    ;
00D43 E417D 4F002   6696              CTL HIO,RSTODID & PSHLD 2
                    6697              LSHFT2 R3                        ;MOVE RECOVERY ACTION INTO POSITION
00D44 01733 887FF   6697 +            ADD AB,LSBZ,R3,R3 & CLPCNT EVCNTZ
00D45 3413E FFFFF   6698              ALDREG R3,IREG                   ;MOVE IT TO I-REG
00D46 370EF FFFFF   6699              RDREGQ IREG                      ;STROBE XFER. REG.
```

Addr Line CDC O.S. -- ENDING ROUTINE

```
00D47 D0FF0 FFFFF  6700            RDXFR SMC,,R0                    ;RECOVERY ACTION TO R0
00D48 3130D FFFFF  6701            OR AB,B,R0,R13                   ;MERGE RECOV. ACTION WITH STATUS
                   6702    ;
00D49 4505E F0200  6703            TSTRBIT R14,200                  ;ALTERNATE TRACK FLAG?
00D4A FFFFF 3354F  6704            BOZ END11                        ;IF NOT, JUMP
00D4B 373FB F00AF  6705            RDRAMB R11,ALTCYL                ;GET THE ALTERNATE CYLINDER...
00D4C 373F0 F00B0  6706            RDRAMB R0,ALTHEAD                ;AND ALTERNATE HEAD...
00D4D 3410F F00B6  6707            ALDRAM R0,CURHD                  ;IN ORDER TO...
                   6708            JSR.P ILDSENS+2                  ;SET UP SENSE BYTES
00D4E E0402 1F202  6708 +          SETPGE ILDSENS+2_SHR_H#B & JSRP ILDSENS+2
                   6709    ;
00D4F 373F7 F00BA  6710    END11:  RDRAMB R7,ILDFLG                 ;GET ILD FLAGS
00D50 45057 F0404  6711            TSTRBIT R7,404                   ;ILD RUN OR DEFECTIVE TRACK?
00D51 34356 33D55  6712            RTOR R5,R6 & BNZ END15           ;IF YES, JUMP; R/W FLAGS TO R6
                   6713    ;
00D52 37350 F0001  6715            RDMSKB   R0,1                    ;$
00D53 451C0 FFFFF  6729            AND DB,FOUTL,HSIMST,R0           ;HAIO SELECTED?
00D54 3439A 33605  6730            RTOR R9,R10 & BOZ END75          ;NO, PROCESS UNEXPECTED E.I.
                   6731    ;
00D55 34365 FFFFF  6732    END15:  RTOR R6,R5                       ;READ/WRITE FLAGS BACK TO R5
00D56 3715E F0100  6733            MOVE MSK2,IREG & NOP H#100       ;SM-IRPT. MASK TO IREG
                   6734            JSR.P WAIT4SM+1                  ;WAIT FOR SM-DONE
00D57 E0403 1F236  6734 +          SETPGE WAIT4SM+1_SHR_H#B & JSRP WAIT4SM+1
                   6736            JSR.P    RSTMDB                  ;$Reset the MDB board
00D58 E0400 1F34B  6736 +          SETPGE RSTMDB_SHR_H#B & JSRP RSTMDB
                   6738            JSR.P CLRHSSM                    ;RESET HSSM BOARDS
00D59 E0400 1F340  6738 +          SETPGE CLRHSSM_SHR_H#B & JSRP CLRHSSM
00D5A 373FA F025F  6739            RDRAMB R10,STATACT               ;RESTORE SAVED STATUS ACTION
00D5B 45057 F0004  6743            TSTRBIT R7,4                     ;CHECK ILD RUNNING
00D5C FFFFF 33D66  6744            BNZ END17                        ;IF RUNNING, JUMP
00D5D 4535F F0002  6745            MSKRBIT R15,2                    ;CLEAR ALL FLAGS EXCEPT TRUNCATION
00D5E 3705F F002A  6746            RDMSKQ 2A                        ;GET MSB'S OF 2-SECOND TIMER
00D5F D0FF9 FFFFF  6747            RDXFR SMC,,R9                    ;SET 2-SECOND DEVICE END...
00D60 35359 F0D4E  6748            SETRBIT R9,H#D4E                 ;DELAY TIMER
00D61 37380 C0001  6752            RDREGB HIOST,R0 & LDCT 1         ;READ HAIO STATUS
00D62 45050 F0044  6753            TSTRBIT R0,44                    ;E.F. OR ADDR. WORD?
00D63 34365 33566  6754            RTOR R6,R5 & BOZ END17           ;IF NOT, CONTINUE; RESTORE R5
                   6755
                   6756            JSR.P CLREF                      ;CLEAR ANY E.F. FUNCT WORD...
00D64 E0403 1F583  6756 +          SETPGE CLREF_SHR_H#B & JSRP CLREF
                   6757            JSR.P CLREF                      ;AND ADDRESS WORD
```

```
 Addr            Line CDC O.S. - ENDING ROUTINE

00D65 E0403 1F583 6757 +          SETPGE CLREF_SHR_H#B & JSRP CLREF
                  6758    ;
00D66 3710F F002C 6759    END17:  RESETRAM SKRETRY                ;CLEAR SEEKERR RETRY COUNT
00D67 FFFFF E0D79 6766            JMP     END50                   ;$continue
                  6772    ;
                  6773    ;     STATUS:  CHANNEL END ONLY
                  6774    ;
00D68 3735A F0004 6775    END20:  RDMSKB R10,4                    ;STATUS ACTION = 04
00D69 4535D F00FF 6776            MSKRBIT R13,H#FF                ;DELETE EXISTING STATUS
00D6A 373F9 F03E4 6777            RDRAMB R9,DLAYDE                ;GET DEVICE END DELAY TIMER
00D6B 370FF F0287 6778            RDRAMQ H#287                    ;GET LSB'S OF E.F. COUNT
00D6C 4605F F000F 6779            TSTQBIT H#F                     ;ISOLATE THE 4 LSB'S
00D6D 6019F F03E4 6780            XOR AQ,FOUTL,R9,RAME & NOP DLAYDE ;CHANGE THE DELAY TIMER AND...
00D6E FFFFF E0D38 6781            JMP END01                       ;SEND CHANNEL END ONLY
                  6782    ;
                  6783    ;   STATUS:  ATTENTION
                  6784    ;
00D6F 3735A F0001 6785    END30:  RDMSKB R10,1                    ;STATUS ACTION = 1
00D70 4535D F10FF 6786            MSKRBIT R13,10FF                ;DELETE PREVIOUS STATUS
00D71 3535D F8000 6787            SETRBIT R13,8000                ;ADD STATUS:  ATTENTION
00D72 34338 E0D39 6791            RTOR R3,R8 & JMP END01+1        ;BIT FOR THIS DRIVE (24-31) TO R8
                  6792    ;
                  6793    ;     STATUS:  DEVICE END only
                  6794    ;
00D73 3735A F0005 6795    END40:  RDMSKB R10,5                    ;STATUS ACTION = 5
00D74 4535D F00FF 6796            MSKRBIT R13,H#FF                ;DELETE EXISTING STATUS
00D75 3535D F0400 6797    END40A: SETRBIT R13,400                 ;SET STATUS: DEVICE END
                  6802    ;
00D76 373F9 F03E4 6803    END41:  RDRAMB R9,DLAYDE                ;GET DEV. END DELAY TIMER
00D77 35359 F0020 6804            SETRBIT R9,20                   ;Make it longer
00D78 373F7 F00BA 6805            RDRAMB R7,ILDFLG                ;GET ILD FLAGS
                  6806    ;
00D79 45057 F0004 6807    END50:  TSTRBIT R7,4                    ;ILD RUNNING?
00D7A E46F1 3357F 6808            CTL HIO, ENCHT & BOZ END50B     ;IF NOT, JUMP
                  6809    ;
00D7B 341EF F01F2 6810            ALDRAM R14,R14SAVL              ;SAVE R14 FOR ILD ENDING
00D7C 6505A F0004 6811            TSTRMSK R10,4                   ;IF STATUS ACTION NOT = 4...
00D7D E4402 13E0F 6815    END50A: CTL HIO,HDRVGTR & CJSR NALUZ,RELDRIV ;RELEASE & DE-SELECT THE DRIVE
00D7E E1008 E0D87 6817            CTL HSSM,RSTUSEL & JMP END51
                  6818    ;
                  6819    END50B: EQU $
```

Addr Line CDC O.S. - ENDING ROUTINE

```
00D7F 370FF F02F4  6835            RDRAMQ TWODRIV                    ;TWO DRIVES PER UNIT?
00D80 0F100 3357D  6836            ADD DZC,FOUTL,ZERO,ZERO & BOZ END50A ;IF NOT, JUST RELEASE DRIVE
                   6837    ;
00D81 4505F F1000  6838            TSTRBIT R15,1000                  ;COLLISION FLAG SET?
00D82 E4402 33D87  6839            CTL HIO,HDRVGTR & BNZ END51       ;YES, ALL THIS STUFF HAS BEEN DONE
                   6840    ;
00D83 3535D F0001  6841            SETRBIT R13,1
00D84 E100B 1F609  6845            CTL HSSM,RSTUSEL & JSR SELREL     ;SELECT & RELEASE UPPER DRIVE
00D85 5535D F0001  6847            RSTRBIT R13,1
00D86 FFFFF 1F609  6851            JSR SELREL                        ;SELECT & RELEASE LOWER DRIVE
                   6853
                   6854    END51:  EQU $
00D87 E1003 FFFFF  6859            CTL HSSM,MPCRST                   ;RESET ANY HSSM OPERATION
00D88 E417D FFFFF  6860            CTL HIO, RSTODID                  ;RESET ODR/IDR
00D89 4505E F0002  6865            TSTRBIT R14,2                     ;CHECK FOR FORCED E.F. FLAG
00D8A 603EE 33DFC  6866            XOR AQ,B,R14,R14 & BNZ END70      ;IF FORCED E.F., JUMP
00D8B 017AA FFFFF  6867            ADD AB,LSBZ,R10,R10               ;MOVE STAT. ACTION INTO POSITION
00D8C 4505F F1000  6868            TSTRBIT R15,1000                  ;COLLISION (ILD ADV. CYL.) FLAG?
00D8D FFFFF 33D94  6869            BNZ END51A                        ;IF SET, JUMP
                   6874    ;
00D8E 4505D F007F  6875            TSTRBIT R13,7F                    ;ISOLATE PHYSICAL DRIVE NO. IN Q...
00D8F 603DD FFFFF  6876            XOR AQ,B,R13,R13                  ;AND DELETE IT FROM R13
00D90 373F2 F02F4  6877            RDRAMB R2,TWODRIV                 ;TWO DRIVES PER UNIT?
00D91 FFFFF 33593  6878            BOZ $+2                           ;IF NOT, JUMP
00D92 334F0 FFFFF  6879            OR ZB,RSQBZ,,R0                   ;1/2 DRIVE NO.
00D93 303DD FFFFF  6880            OR AQ,B,R13,R13                   ;LOGICAL DRIVE TO R13
                   6881    ;
00D94 45057 F0004  6882    END51A: TSTRBIT R7,4                      ;CHECK ILD RUNNING
                   6883            BNZ.P ILDEND                      ;IF RUNNING, RETURN TO ILD
00D95 E0402 338C2  6883 +          SETPGE ILDEND_SHR_H#B & BNZP ILDEND
                   6884    ;
00D96 5535F F1000  6885            RSTRBIT R15,1000                  ;RESET COLLISION FLAG
00D97 4505F F0002  6889            TSTRBIT R15,2                     ;CHECK TRUNCATION FLAG
00D98 603FF 3359A  6890            XOR AQ,B,R15,R15 & BOZ END51B     ;IF NOT, JUMP; IF YES, RESET IT
00D99 3535A F0080  6891            SETRBIT R10,80                    ;SET CHANNEL TRUNCATION BIT
```

Addr Line CDC O.S. - ENDING ROUTINE

```
                    6892    ;
00D9A 741DA FFFFF   6893    END51B:   NALDREG R13,UBR1                ;LOAD THE...
00D9B 749DB FFFFF   6894              NALDREGU R13,UBR2               ;STATUS TO THE...
00D9C 741AC FFFFF   6895              NALDREG R10,UBR3                ;U-BUS REGISTER
                    6896    ;
00D9D 37351 F0005   6897              RDMSKB R1,5                     ;GET INTERRUPT MASK
00D9E 45081 FFFFF   6898              AND DB,Q,HIOST,R1               ;CHECK FOR E.F. OR PARITY ERR.
00D9F E0204 33DD0   6899              CTL SMC,RSTIDAK & BNZ END58     ;IF E.F. OR PARITY ERR., JUMP
00DA0 E4401 FFFFF   6900    END52:    CTL HIO,HDRVGTS                 ;ENABLE STATUS TO HIO
00DA1 37351 F1004   6902              RDMSKB   R1,1004                ;$Mask for IA/EF
00DA2 E417B FFFFF   6906              CTL HIO,SETEI                   ;SET E.I.
                    6907    ;
00DA3 3735B F001E   6908    END52A:   RDMSKB R11,1E
00DA4 D0FFB FFFFF   6909              RDXFR SMC,,R11                  ;SET 50-ms...
00DA5 3535B F0848   6910              SETRBIT R11,848                 ;E.I. TIMER TO R11
00DA6 45081 335CA   6911              AND DB,Q,HIOST,R1 & BOZ END56   ;IF 50-ms. TIME-OUT, JUMP
00DA7 133FB 335A6   6912              DECRB R11 & BOZ $-1             ;IF NO INTERRUPT, JUMP BACK
                    6913    ;
00DA8 46150 F0004   6914              TESTQ 4                         ;CHECK FOR E.F.
00DA9 343D0 33DCD   6915              RTOR R13,R0 & BNZ END57         ;IF E.F., JUMP
                    6936    ;
                    6937    ;     I.A. RECEIVED.  DELAY AND CHECK FOR AN E.F.
                    6938    ;
00DAA 420FF 4F004   6939    END52B:   RESETQ & PSHLD 4                ;2 µsec TIMER
00DAB 3708F B3DAF   6940              RDREGQ HIOST & CJPUP NALUZ,END53+1 ;IF E.F., DON'T DE-SELECT HAIO
00DAC 4604F 88004   6941              AND DQ,Q,MSK1 & CLPCNT EVCNTZ,H#4 ;CHECK FOR E.F.; DECR. TIMER
00DAD E0204 33DAF   6942              CTL SMC,RSTIDAK & BNZ END53+1   ;IF E.F., DON'T DE-SELECT HAIO
                    6943
00DAE E0090 FFFFF   6944    END53:    CTL SMC,UMPRST                  ;DE-SELECT THE HIO
00DAF 343A3 E2064   6945              RTOR R10,R3   & JMP EITRACE     ;STATUS 12 MSB'S TO R3 (TRACE OPT.)
00DB0 3715E F02A0   6946    END53A:   MOVE MSK2,IREG & NOP H#2A0      ;LOC. OF E.I. DISPLAY
                    6947              JSR.P AGE01                     ;UPDATE E.I. DISPLAY
00DB1 E0400 1F35B   6947 +            SETPGE AGE01_SHR_H#B & JSRP AGE01
00DB2 5535F FEFFF   6948              RSTRBIT R15,H#EFFF              ;CLEAR FLAGS EXCEPT COLLISION
00DB3 37351 F000F   6964              RDMSKB R1,H#F                   ;Allow E.F's on all HAIO's
00DB4 3411F F002D   6965              ALDRAM R1,EFALLOW               ;Save bits for E.F's allowed
                    6966    ;
                    6967    END53C:   EQU $
00DB5 5504D F00FF   6973              MASK DB,Q,MSK1,R13 & NOP H#FF   ;DELETE DRIVE NUMBER
00DB6 66150 F8000   6974              TSTQEQU  8000                   ;CHECK FOR ATTN./±BUSY
00DB7 E0084 335E6   6975              CTL SMC,SMCRST & BOZ END62      ;IF TRUE, JUMP
00DB8 4605F F0C00   6976              TSTQBIT H#C00                   ;ISOLATE CE/DE ONLY
00DB9 66150 F0400   6977              TSTQEQU  400                    ;CHECK DEVICE END ONLY
00DBA E4402 33DBE   6978              CTL HIO,HDRVGTR & BNZ END54     ;IF NOT, JUMP
                    6979    ;
```

```
 Addr            Line CDC O.S. - ENDING ROUTINE

00DBB 343D0 1F329  6982            RTOR R13,R0 & JSR SEL40          ;COMPUTE BIT FOR THIS DRIVE
00DBC FFFFF 1F1D5  6984            JSR SEL07                        ;CLEAR SEEK COMP. PEND. THIS DR.
                   6985            JSR.P CLRSECT                    ;CLEAR SAVED SECTOR THIS DRIVE
00DBD E0403 1F367  6985 +          SETPGE CLRSECT_SHR_H#B & JSRP CLRSECT
                   6986    ;
00DBE 373F1 F0271  6987    END54:  RDRAMB R1,EFSTACK+1              ;GET CURRENT COMMAND CODE
00DBF 65051 F0100  6988            TSTRMSK R1,100                   ;CHECK FOR SENSE I/O
00DC0 FFFFF 33DC3  6989            BNZ END55                        ;If not, continue
00DC1 4505D F0200  6990            TSTRBIT R13,200                  ;Status = Unit Check?
00DC2 FFFFF 1352E  6991            CJSR ALUZ,CLRSENS                ;If not, clear Sense Bytes
                   6992    ;
00DC3 4505F F1000  6993    END55:  TSTRBIT R15,1000                 ;CHECK COLLISION FLAG
00DC4 E7C00 33DDF  6994            CTL OCP,OCPDSEL & BNZ END61      ;IF SET, SEND COLLISION STATUS
00DC5 4505D F6200  6999            TSTRBIT R13,6200                 ;COLLISION STATUS OR UNIT CHECK?
00DC6 E4402 335EF  7003            CTL HIO,HDRVGTR & BOZ END64      ;IF NOT, jump; disable U-bus
00DC7 373F1 F002E  7005            RDRAMB R1,BITHIO                 ;GET THE BIT FOR THIS HAIO
00DC8 3411F F002D  7006            ALDRAM R1,EFALLOW                ;NEXT E.F. MUST BE ON THIS HAIO
00DC9 FFFFF E00D2  7007    END55A: JMP WAIT00                       ;RETURN TO WAIT LOOP
                   7008    ;
00DCA E4177 FFFFF  7009    END56:  CTL HIO,RSTEI                    ;RESET E.I.
00DCB 3535A F0700  7010            SETRBIT R10,700                  ;Add bits 32-34 for E.I. time-out
00DCC 343D0 E0DAE  7011            RTOR R13,R0 & JMP END53          ;UPDATE STATUS DISPLAY
                   7012
                   7013    ;     E.F. RECEIVED FOLLOWING AN E.I.
                   7014    ;
00DCD 37351 F0002  7015    END57:  RDMSKB R1,2                      ;MASK FOR E.I. SENT
00DCE 45081 FFFFF  7016            AND DB,Q,HIOST,R1                ;CHECK FOR E.I. SENT
00DCF 3708F 33DDC  7017            RDREGQ HIOST & BNZ END60         ;IF SENT, WAIT FOR I.A.
                   7018    ;
00DD0 4605F F0001  7019    END58:  TSTQBIT 1                        ;CHECK U-BUS PARITY
00DD1 FFFFF 335D4  7020            BOZ $+3                          ;IF NOT, JUMP
                   7021            JSR.P CLREF                      ;CLEAR ANY E.F. AND...
00DD2 E0403 1F583  7021 +          SETPGE CLREF_SHR_H#B & JSRP CLREF
00DD3 FFFFF E0DA0  7022            JMP END52                        ;FORCE AN E.I.
                   7023    ;
00DD4 4505D F8000  7028            TSTRBIT R13,8000                 ;CHECK FOR ATTENTION STATUS
00DD5 E4402 33DDA  7029            CTL HIO,HDRVGTR & BNZ END59      ;IF ATTENTION, NOT A FORCED E.F.
00DD6 4505D F0C00  7030            TSTRBIT R13,H#C00                ;ISOLATE CHAN.END/DEV. END
00DD7 6605F F0400  7031            TSTQMSK 400                      ;CHECK FOR DEVICE END ONLY
00DD8 FFFFF 335DA  7032            BOZ END59                        ;IF D.E. ONLY, NOT A FORCED E.F.
00DD9 3735E F0002  7033            RDMSKB R14,2                     ;SET FORCED E.F. FLAG
                   7034    ;
```

Addr Line CDC O.S. - ENDING ROUTINE

```
                    7039    END59:    LDAR.P EFPROC3
00DDA E0400 5F07D   7039 +            SETPGE EFPROC3_SHR_M#B & LDARP EFPROC3
00DDB E0088 E0002   7041              CTL SMC,WMARST & JMP WAIT00        ;PROCESS THE E.F.
                    7042    ;
00DDC 3535F F1000   7043    END60:    SETRBIT R15,1000                   ;SET COLLISION FLAG
00DDD 37351 F1000   7044              RDMSKB R1,1000                     ;ALLOW I.A. ONLY
00DDE FFFFF E0DA3   7045              JMP END52A                         ;WAIT FOR IT
                    7046
                    7047    ;   E.F./E.I. COLLISION
                    7048    ;
00DDF 373F1 F002E   7049    END61:    RDRAMB R1,BITHIO                   ;GET ALLOWED E.F. BIT
00DE0 3411F F002D   7050              ALDRAM R1,EFALLOW                  ;NEXT E.F. MUST BE ON THIS HAIO
00DE1 E4402 5F5E3   7051              CTL HIO,HDRVGTR & LDAR $+D#2
00DE2 FFFFF E0063   7052              JMP EFPROC1                        ;GET THE E.F. FOR DISPLAY
00DE3 3535D F7000   7053              SETRBIT R13,7000                   ;STAT. MOD., C.U. END, BUSY
00DE4 3735A F0003   7054              RDMSKB R10,3                       ;STATUS ACTION = 03
00DE5 E4177 E0D76   7055              CTL HIO,RSTEI & JMP END41          ;RESET E.I. SENT STATUS & SEND E.I.
                    7056    ;
00DE6 373F1 F002F   7057    END62:    RDRAMB R1,ATTNHIO                  ;GET ATTN. HIO NUMBER
00DE7 34712 FFFFF   7058              OR AZ,LSBZ,R1,R2                   ;COMPUTE LOCATION...
00DE8 01712 FFFFF   7059              ADD AB,LSBZ,R1,R2                  ;OF ATTENTION PENDING...
00DE9 05342 F0072   7060              ADD DB,B,MSK1,R2 & NOP ATTN1       ;FOR THIS HIO
00DEA 3412E 1F0CD   7061              ALDREG R2,IREG & JSR SMBRDA        ;GET ATTN. PENDING THIS HIO
00DEB 51340 FFFFF   7062              MASK AB,B,R4,R0                    ;DELETE THE BIT...
00DEC 5008F FFFFF   7063              MASK AQ,Q,R8                       ;FOR THIS DRIVE...
00DED 3412E 1F0CE   7064              ALDREG R2,IREG & JSR SMBWRTA       ;ON THIS HIO...
00DEE E4402 E0DC3   7065              CTL HIO,HDRVGTR & JMP END55        ;AND EXIT
                    7066    ;
                    7067    ;   The following code helps to balance system performance when running
                    7068    ;with a fast host and/or Multi-host.
                    7069    ;
00DEF 6505A F0018   7070    END64:    TSTRMSK R10,18                     ;Status Action = 6?
00DF0 FFFFF 33DFB   7071              BNZ END65                          ;If not, go to Wait Loop
00DF1 37352 F0054   7072              RDMSKB R2,SKPEND
00DF2 3412E 1F0CD   7073              ALDREG R2,IREG & JSR SMBRDA        ;Any Seek Complete Pending?
00DF3 37190 335FB   7074              MOVE SPIST,ZERO & BOZ END65        ;If not, go to Wait Loop
00DF4 34304 335F8   7075              RTOR R0,R4 & BOZ END65             ;If no E.F., go to Wait Loop
00DF5 3715E F02B5   7076              MOVE MSK2,IREG & NOP DRONCYL
00DF6 323F3 1F0CD   7077              QTOR R3 & JSR SMBRDA               ;Any drives On-cylinder yet?
00DF7 51104 333AA   7078              MASK AB,FOUTL,R0,R4 & BOZ POLL10   ;If not, Wait Loop via Polling
00DF8 323F1 33BAA   7079              QTOR R1 & BNZ POLL10               ;If any (0-23) *On-cyl., Polling
00DF9 51113 FFFFF   7080              MASK AB,FOUTL,R1,R3                ;If any (24-31) not On-cyl...
00DFA FFFFF 33BAA   7081              BNZ POLL10                         ;Wait loop via Polling routine
00DFB E7C00 E0002   7082    END65:    CTL OCP,OCPDSEL & JMP WAIT00       ;De-select OCP & go to Wait Loop
                    7083
                    7084    ;   FORCED E.F. PROCESSING
                    7085    ;
                    7086    END70:    ADDMBIT 80,SEN00                   ;Add Comm. Reject to Sense 0
00DFC 373F0 F0000   7086 +            RDRAMB  R0,SEN00
00DFD 35350 F0080   7086 +            SETRBIT R0,80
00DFE 3410F F0000   7086 +            ALDRAM  R0,SEN00
```

Addr Line CDC O.S. - ENDING ROUTINE

```
                 7087            SETMBIT 9,SEN07                 ;Sense 7 = Term. by Forced E.F.
00DFF 37350 F0009 7087 +          RDMSKB   R0,9
00E00 3410F F0007 7087 +          ALDRAM   R0,SEN07
00E01 37359 F000F 7088            RDMSKB R9,H#F                   ;STATUS ACTION = 0F
00E02 4535D F00FF 7089            MSKRBIT R13,H#FF                ;DELETE EXISTING STATUS
00E03 37353 F0004 7090            RDMSKB R3,4                     ;RECOVERY ACTION = 4
00E04 FFFFF E0D40 7091            JMP END10A                      ;SEND UNIT CHECK
                 7092
                 7093
                 7094    ;NOTE:  THE FOLLOWING CODE IS EXECUTED ONLY IF THE PROGRAM ATTEMPTS
                 7095    ;TO SEND AN UNEXPECTED E.I.  NO E.I. WILL BE SENT, BUT STATUS WILL BE
                 7096    ;DISPLAYED.
                 7097    ;
00E05 373F9 F03E4 7098    END75:    RDRAMB R9,DLAYDE                ;ALLOW E.F.'S ON ALL HIO'S
00E06 741DA FFFFF 7099            NALDREG R13,UBR1                ;LOAD ERROR STATUS...
00E07 749DB 1F609 7100            NALDREGU R13,UBR2 & JSR SELREL  ;TO THE U-BUS
00E08 433FF E0DCA 7101            RESETR R15 & JMP END56          ;SAME DISPLAY AS UNACKNOWLEDGED...
                 7102    ;                                        E.I.
                 7103
                 7104    ;    THIS SUBROUTINE SELECTS A DRIVE SO THAT IT MAY BE RELEASED.
                 7105    ;UPON ENTRY:  R13 = PHYSICAL DRIVE NO.
                 7106    ;REGISTERS USED:  R0 & Q.
                 7107    ;
00E09 4505D F007F 7116    SELREL:   TSTRBIT R13,7F                  ;ISOLATE DRIVE NO. ONLY
00E0A B2200 F0000 7117            WRTE HSSMA,LDDRBO & NOP 0       ;CLEAR DRIVE BUS OUT REG.
00E0B 323F0 FFFFF 7118            QTOR R0                         ;DRIVE NO. TO R0
00E0C 91E0F FFFFF 7119            WRTL HSSM,LDDRUSEL,R0           ;LOAD UNIT SELECT REG.
00E0D FFFFF 4F007 7120            PSHLD 7
00E0E E1007 887FF 7121            CTL HSSM,SETUSEL & CLPCNT EVCNTZ  ;SELECT THE DRIVE
                 7122    ;
                 7123    ;    RELEASE & DE-SELECT A DRIVE.
                 7124    ;
00E0F B2200 F0200 7125    RELDRIV:  WRTE HSSMA,LDDRBO & NOP H#200   ;BUS OUT TO RELEASE THE DRIVE
00E10 FFFFF C0006 7126            LDCT 6
00E11 E100A 98611 7127            CTL HSSM,SETAG3 & CJPCNT EVCNTZ,$  ;RELEASE THE DRIVE AND...
00E12 E100B FFFFF 7131            CTL HSSM,RSTUSEL                ;DE-SELECT IT
00E13 E1016 FFFFF 7132            CTL HSSM,RSTAG3
00E14 B2200 AF000 7133            WRTE HSSMA,LDDRBO & RTS 0       ;CLEAR DATA BUS OUT & RETURN
                 7134    ;
                 7144
                 7145    TITLE OCP HANDLER
```

Addr Line OCP HANDLER

```
                    7147    ;
                    7148    ; THIS SUBROUTINE SELECTS THE OCP IN A CHAN A/B ENVIRONMENT
                    7149    ;
                    7150    ; ENTRY PARAMETERS:    NONE
                    7151    ; EXIT  PARAMETERS:    ALU = 0, OCP SELECTED.
                    7152    ;                      ALU = 1, OCP NOT SELECTED
                    7153    ; REGISTERS USED:  R0
                    7154    ;
00E15 3410E FFFFF   7155    SELOCP:   ALDREG R0,IREG                    ;LOAD RETURN ADDRESS
00E16 34909 C0FFF   7156              ALDREGU R0,IREGH & LDCT H#FFF
00E17 E7000 FFFFF   7157    SELOCPA:  CTL OCP,OCPREQ                    ;REQUEST THE OCP
00E18 C7F00 FFFFF   7158              RD OCP,OCPSTAT,R0
00E19 C7F00 FFFFF   7159              RD OCP,OCPSTAT,R0
00E1A C7F00 FFFFF   7160              RD OCP,OCPSTAT,R0
00E1B 45150 F0001   7161              TSTBIT R0,1                       ;CHECK FOR OCP AVAILABLE
00E1C FFFFF 33E22   7162              BNZ SELOCPB                       ;IF IT IS, JUMP
00E1D FFFFF 98617   7163              CJPCNT EVCNTZ,SELOCPA             ;ELSE DECREMENT TIMER
                    7164    ;
00E1E E7C00 FFFFF   7165              CTL OCP,OCPDSEL                   ;CLEAR OCP REQUEST
00E1F 3705F F0020   7166              RDMSKQ 20                         ;OCP NOT AVAILABLE SENSE
                    7167              JSR.P SETSEN12                    ;SET SENSE BYTE 12
00E20 E0403 1F243   7167 +            SETPGE SETSEN12_SHR_H#B & JSRP SETSEN12
00E21 0F160 6F7FF   7168              ADD DZC,FOUTL,SMCST,R0 & JIR      ;ALU NOT = 0; OCP NOT AVAIL.
                    7169    ;
00E22 E7800 FFFFF   7170    SELOCPB:  CTL OCP,OCPRST                    ;RESET ANY OCP PARITY AND...
00E23 47160 6F7FF   7171              AND DZ,FOUTL,SMCST,ZERO & JIR     ;TAKE NORMAL RETURN (ALU = 0)
```

```
Addr            Line OCP HANDLER


                7173    ;
                7174    ; THIS ROUTINE WILL ISSUE COMMANDS TO THE OCP
                7175    ;
                7176    ;       ENTRY PARAMETERS            Q = COMMAND (1-11) HEX
                7177    ;                                   R0 = PARAMETER #1
                7178    ;                                   R1 = PARAMETER #2
                7179    ;
                7180    ;       EXIT PARAMETERS             SAME AS R0 AND R1
                7181    ; REGISTERS USED:  R0-R2 & Q.
                7182    ;
00E24 321FF F0260  7183 ISSCMD:   QLDRAM COMMAND                    ;SAVE INCOMMING COMMAND
00E25 3705F F186A  7184           RDMSKQ H#186A                     ;SET 10-MS. TIMER
00E26 E7000 4F002  7185           CTL OCP,OCPREQ & PSHLD 2          ;REQUEST THE OCP
00E27 C7F02 887FF  7186           RD OCP,OCPSTAT,R2 & CLPCNT EVCNTZ ;GET OCPSTATUS
00E28 45352 F0401  7187           MSKRBIT R2,401                    ;ISOLATE ACTIVE & AVAIL
00E29 65152 F0001  7188           TSTREQU R2,1                      ;NOT ACTIVE AND AVAIL ?
00E2A 120FF 3362D  7189           DECRQ & BOZ ISSCMD4$              ;IF YES, JUMP; DECR. TIMER
00E2B FFFFF 33E26  7190           BNZ ISSCMD+2                      ;IF NOT TIME-OUT, CHECK AGAIN
00E2C 37160 E0E43  7191           MOVE SMCST,ZERO & JMP ISSRTN      ;10-MS. TIME-OUT RETURN
                7192    ;
00E2D B7000 F0023  7193 ISSCMD4:  WRTE OCP,OCPADR & NOP D#35        ;Loc. of Parameter 1 lsb's
00E2E 9710F 4F001  7194           WRTL OCP,OCPDATA,R0 & PSHLD 1     ;STORE PARAMETER #1 LSB'S
00E2F 9710F 887FF  7195           WRTL OCP,OCPDATA,R0 & CLPCNT EVCNTZ
00E30 B7000 F0022  7196           WRTE OCP,OCPADR & NOP D#34        ;Loc. of Parameter 1 msb's
00E31 01700 FFFFF  7197           ADD     AB,LSBZ,R0,R0             ;Move msb's...
00E32 01700 4F002  7198           ADD     AB,LSBZ,R0,R0 & PSHLD 2   ;into position
00E33 A710F 887FF  7199           WRTU OCP,OCPDATA,R0 & CLPCNT EVCNTZ ;STORE PARAMETER #1 MSB'S
                7200    ;
00E34 B7000 F0025  7201           WRTE OCP,OCPADR & NOP D#37        ;Loc. of Parameter 2 lsb's
00E35 9711F 4F001  7202           WRTL OCP,OCPDATA,R1 & PSHLD 1     ;STORE PARAMETER #2 LSB'S
00E36 9711F 887FF  7203           WRTL OCP,OCPDATA,R1 & CLPCNT EVCNTZ
00E37 B7000 F0024  7204           WRTE OCP,OCPADR & NOP D#36        ;Loc. of Parameter 2 msb's
00E38 01711 FFFFF  7205           ADD     AB,LSBZ,R1,R1             ;Move msb's...
00E39 01711 4F002  7206           ADD     AB,LSBZ,R1,R1 & PSHLD 2   ;into position
00E3A A711F 887FF  7207           WRTU OCP,OCPDATA,R1 & CLPCNT EVCNTZ ;STORE PARAMETER #2 MSB'S
                7208    ;
00E3B B7000 F0021  7209           WRTE OCP,OCPADR & NOP D#33        ;Loc. of OCP command
00E3C 373F1 F0260  7210           RDRAMB R1,COMMAND                 ;Get Ucp command and...
00E3D 9711F 4F001  7211           WRTL OCP,OCPDATA,R1 & PSHLD 1     ;load it
00E3E 9711F 887FF  7212           WRTL OCP,OCPDATA,R1 & CLPCNT EVCNTZ
00E3F FFFFF 1F65A  7213           JSR     OCPST                     ;Check for parity errors
                7214    ;
00E40 E7140 FFFFF  7215           CTL OCP,SCNDACT                   ;SET COMMAND ACTIVE
00E41 45051 F0180  7216           TSTRBIT R1,180                    ;CHECK FOR LONG COMMAND
00E42 37160 33E44  7217           MOVE SMCST,ZERO & BNZ $+2         ;IF YES, JUMP
00E43 E7C00 AF7FF  7218 ISSRTN:   CTL OCP,OCPDSEL & RTS             ;DE-SELECT OCP & RETURN
                7219    ;
00E44 45051 F0080  7220           TSTRBIT R1,80                     ;CHECK FOR LONG COMMAND
00E45 FFFFF 33E48  7221           BNZ ISSCMD3$                      ;JUMP IF LONG
00E46 FFFFF 1F654  7222           JSR GETOCPST$                     ;GET OCP STATUS
00E47 45050 F1400  7223           TSTRBIT R0,1400                   ;ISOLATE SHORT DONE & CMDACT
00E48 6605F F1000  7224           TSTQMSK 1000                      ;SHORT DONE ONLY?
00E49 FFFFF 33E46  7225           BNZ $-3                           ;IF NOT, WAIT FOR IT
00E4A 37160 AF7FF  7226           MOVE SMCST,ZERO & RTS             ;YES, RETURN; RESET XMC PARITY
```

Addr Line OCP HANDLER

```
                    7227    ;
00E4B FFFFF 1F654   7228    ISSCMD3:  JSR GETOCPST$                    ;GET OCP STATUS
00E4C 45051 F0010   7229              TSTRBIT R1,10                    ;CHECK FOR OCP PANEL "CLEAR"
                    7230              CJPOP.P NALUZ,ILD28              ;IF YES, TERMINATE ILD
00E4D E0402 B3933   7230 +            SETPGE ILD28_SHR_H#B & CJPOPP NALUZ,ILD28
00E4E 45050 F9400   7231              TSTRBIT R0,9400                  ;LONG & SHORT DONE & ACTIVE
00E4F 6605F F9000   7232              TSTMSK 9000                      ;LONG DONE AND SHORT DONE ONLY?
00E50 37160 33661   7233              MOVE SMCST,ZERO & BOZ GETPARAM$  ;JUMP IF DONE
                    7234    ;
00E51 37352 F0E91   7235              RDMSKB R2,ISSCMDC                ;RETURN LOC. FROM THE WAIT LOPP
00E52 E7C00 1F69B   7236              CTL OCP,OCPDSEL & JSR SAVRET     ;SAVE IT AND...
                    7237              JPOP.P    WAIT100A               ;GO TO THE WAIT LOOP
00E53 E0402 BF017   7237 +            SETPGE WAIT100A_SHR_H#B & JPOPP WAIT100A
                    7238    ;
00E54 B7000 F0027   7239    GETOCPST: WRTE OCP,OCPADR & NOP H#27
00E55 C7F11 4F001   7240              RD OCP,OCPDATA,R1 & PSHLD 1
00E56 C7F11 887FF   7241              RD OCP,OCPDATA,R1 & CLPCNT EVCNTZ ;GET STATUS MSB'S
00E57 D0FF0 FFFFF   7242              RDXFR SMC,,R0
00E58 3400F 1F65A   7243              RTOQ R0 & JSR OCPST$             ;STATUS MSB'S TO Q & CHECK PARITY
00E59 30300 AF7FF   7244              OR AQ,B,R0,R0 & RTS              ;MERGE MSB'S & RETURN
                    7245    ;
00E5A C7F00 C0001   7246    OCPST:    RD OCP,OCPSTAT,R0 & LDCT 1
00E5B C7F00 9865B   7247              RD OCP,OCPSTAT,R0 & CJPCNT EVCNTZ,$ ;READ OCP STATUS
00E5C 45150 F0006   7248              TSTBIT R0,6                      ;ANY PARITY ERRORS?
00E5D FFFFF A37FF   7249              CRTS ALUZ                        ;NO, RETURN
                    7250    ;
00E5E 3704F 8F040   7251    OCPST1:   RDREGQ MSK1 & POP H#40           ;OCP PARITY ERROR SENSE
                    7252              JSR.P SETSEN12                   ;SAVE IT IN BYTE 12
00E5F E0403 1F243   7252 +            SETPGE SETSEN12_SHR_H#B & JSRP SETSEN12
00E60 E7C00 E04AD   7253              CTL OCP,OCPDSEL & JMP READ62     ;SEND EQUIP. CHECK
                    7254
00E61 45050 F2000   7255    GETPARAM: TSTRBIT R0,2000                  ;CHECK PARAMETER #1 AVAIL.
00E62 34301 33E64   7256              RTOR R0,R1 & BNZ $+2             ;IF YES, GET THE DATA
00E63 E7C00 BF683   7257              CTL OCP,OCPDSEL & JPOP ISSCMDA2$ ;NOT AVAILABLE, RE-ISSUE COMMAND
00E64 3710F F00A6   7258              RESETRAM PARCNT                  ;RESET PARITY RETRY COUNT
00E65 FFFFF 5F666   7259              LDAR $+1                         ;PARITY ERROR RETRY LOC.
00E66 B7000 F0022   7260              WRTE OCP,OCPADR & NOP D#34       ;Loc. of Parameter 1 msb's
00E67 C7F10 4F001   7261              RD OCP,OCPDATA,R0 & PSHLD 1
00E68 C7F10 887FF   7262              RD OCP,OCPDATA,R0 & CLPCNT EVCNTZ ;GET PARAMETER #1 MSB'S AND...
00E69 01700 4F002   7263              ADD AB,LSBZ,R0,R0 & PSHLD 2
                    7264              LSHFT2 R0                        ;MOVE THEM INTO POSITION
00E6A 01700 887FF   7264 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
00E6B B7000 F0023   7265              WRTE OCP,OCPADR & NOP D#35       ;Loc. of Parameter 1 lsb's
00E6C 34301 C0002   7266              RTOR R0,R1 & LDCT 2              ;Parameter 1 msb's to R1
00E6D C7F10 9866D   7267              RD OCP,OCPDATA,R0 & CJPCNT EVCNTZ,$ ;GET PARAMETER #1 LSB'S
                    7268              JSR.P XMCTPAR                    ;CHECK CT-BUS PARITY
00E6E E0403 1F35B   7268 +            SETPGE XMCTPAR_SHR_H#B & JSRP XMCTPAR
00E6F 31310 33E76   7269              OR AB,B,R1,R0 & BNZ OCPCT$       ;IF ERROR, JUMP; BOTH BYTES TO R0
00E70 C7F01 4F001   7270    OCPPAR:   RD OCP,OCPSTAT,R1 & PSHLD 1
00E71 C7F01 887FF   7271              RD OCP,OCPSTAT,R1 & CLPCNT EVCNTZ ;READ OCP STATUS
```

```
 Addr            Line OCP HANDLER

00E72 45151 F0100  7272            TSTBIT R1,100                    ;OCP MEMORY PARITY?
00E73 E7800 33643  7273            CTL OCP,OCPRST & BOZ ISSRTN$     ;IF NOT, NORMAL RETURN
                   7274            JSR.P XMCTPAR1                   ;POSSIBLE RETRY
00E74 E0403 1F35E  7274 +          SETPGE XMCTPAR1_SHR_H#B & JSRP XMCTPAR1
00E75 FFFFF E0E5E  7275            JMP OCPST1                       ;RETRIES FAILED; EQUIP. CHECK
                   7276    ;
00E76 37353 F0004  7277    OCPCT:  RDMSKB R3,4                      ;SET CT-BUS PARITY STATUS
00E77 E7C00 FFFFF  7278            CTL OCP,OCPDSEL                  ;DE-SELECT THE OCP
00E78 3705F F0040  7279            RDMSKQ 40                        ;OCP PARITY SENSE
                   7280            JSR.P SETSEN12                   ;SET SENSE BYTE 12
00E79 E0403 1F243  7280 +          SETPGE SETSEN12_SHR_H#B & JSRP SETSEN12
00E7A FFFFF E03BA  7281            JMP IRPT12+1                     ;PROCESS THE ERROR
                   7282
                   7283    TITLE ILD  MISC. ROUTINES
                   7284
                   7285
                   7286    ;    THE FOLLOWING ROUTINE MUST BE USED WHEN ISSUING COMMANDS TO THE OCP
                   7287    ;WHEN RUNNING ILD'S.
                   7288    ;    IT IS ENTERED WITH:  CALL R2,ISSCMDA (OR ISSCMDA1)
                   7289    ;
00E7B 3705F F0001  7290    ISSCMDA1: RDMSKQ 1                       ;COMMAND TO PRINT MESSAGE
00E7C 3492F F01F8  7291    ISSCMDA:  ALDRAMU R2,RETURN1A            ;SAVE ILD RETURN
00E7D 3412F F01EC  7292              ALDRAM R2,RETURN1
00E7E 321FF F01F5  7293              QLDRAM OCPCMD                  ;SAVE OCP COMMAND
00E7F 3410F F01F6  7294              ALDRAM R0,PARAM1               ;SAVE PARAM. 1
00E80 3411F F0066  7295              ALDRAM R1,PARAM2               ;SAVE PARAM. 2
00E81 37352 F1000  7296              RDMSKB R2,WAIT101              ;RETURN LOC. FROM WAIT LOOP
00E82 FFFFF 1F69B  7297              JSR SAVRET$                    ;SAVE IT
00E83 3705F F0026  7298    ISSCMDA2: RDMSKQ H#26                    ;100 µsec. COMMAND ACTIVE TIMER
                   7299    ;
00E84 0F320 E0E15  7300    ISSCMDB:  CALL R0,SELOCP                 ;SELECT THE OCP
                   7301              BNZ.P    WAIT100A              ;IF NOT SELECTED, WAIT LOOP
00E85 E0402 33817  7301 +          SETPGE WAIT100A_SHR_H#B & BNZP WAIT100A
00E86 45150 F0400  7302              TSTBIT R0,400                  ;CHECK COMMAND ACTIVE
00E87 120FF 3368A  7303              DECRQ & BOZ $+3                ;IF NOT, CONTINUE; DECR. TIMER
00E88 FFFFF 33E84  7304              BNZ ISSCMDB$                   ;IF NOT TIME-OUT, CHECK AGAIN
00E89 E7C00 E1017  7305              CTL OCP,OCPDSEL & JMP WAIT100A ;COMMAND ACTIVE, RETURN TO WAIT
                   7306    ;
00E8A 370FF F01F5  7307              RDRAMQ OCPCMD                  ;RETRIEVE OCP COMMAND
00E8B 373F0 F01F6  7308              RDRAMB R0,PARAM1               ;RETRIEVE PARAM. 1
00E8C 373F1 F0066  7309              RDRAMB R1,PARAM2               ;RETRIEVE PARAM. 2
00E8D FFFFF 1F624  7310              JSR ISSCMD$                    ;ISSUE OCP COMMAND
                   7311    ;
00E8E 371FE F01EC  7312    RTRN1:    MOVE RAME,IREG & NOP RETURN1   ;GET THE RETURN...
00E8F 371F9 F01F8  7313              MOVE RAME,IREGH & NOP RETURN1A ;LOCATION AND...
00E90 333F0 6F7FF  7314              OR ZB,B,,R0 & JIR              ;RETURN TO THE ILD
                   7315    ;
                   7316    ;    THE PROGRAM WILL ENTER THE FOLLOWING ROUTINE FROM THE WAIT LOOP WHILE
                   7317    ;WAITING FOR AN OPERATOR RESPONSE TO AN OCP MESSAGE.
                   7318    ;
00E91 3705F F0026  7319    ISSCMDC:  RDMSKQ H#26                    ;100 US. COMMAND ACTIVE TIMER
00E92 0F320 E0E15  7320              CALL R0,SELOCP                 ;SELECT THE OCP
```

Addr Line ILD MISC. ROUTINES

```
                  7321              BNZ.P WAIT00                    ;IF NOT SELECTED, WAIT LOOP
00E93 E0400 33802 7321 +            SETPGE WAIT00_SHR_H#B & BNZP WAIT00
00E94 15150 F0400 7322              TSTBIT R0,400                   ;CHECK COMMAND ACTIVE
00E95 120FF 33698 7323              DECRQ & BOZ $+3                 ;DECR. TIMER;  IF NOT ACTIVE, JUMP
00E96 FFFFF 33E92 7324              BNZ ISSCMDC+1                   ;IF NOT TIME-OUT, CONTINUE WAITING
00E97 E7C00 E0002 7325              CTL OCP,OCPDSEL & JMP WAIT00    ;COMMAND ACTIVE, RETURN TO WAIT
                  7326    ;
                  7327              JSR.P    WAIT103                ;RESTORE REGISTERS
00E98 E0402 1F00A 7327 +            SETPGE WAIT103_SHR_H#B & JSRP WAIT103
00E99 FFFFF 1F64B 7328              JSR ISSCMD3$                    ;CHECK FOR OPERATOR RESPONSE
00E9A FFFFF E0E8E 7329              JMP RTRN1                       ;RETURN TO ILD
                  7330    ;
00E9B 3412F F00BF 7331    SAVRET:   ALDRAM R2,RETURN                ;SAVE WAIT LOOP RETURN LSB'S
00E9C 3492F AF0BE 7332              ALDREGU R2,RAME & RTS RETURNA   ;SAVE WAIT LOOP RETURN MSB'S
```

Addr Line ILD MISC. ROUTINES

```
01000              7334    ORG H#1000
                   7335    ;
01000 FFFF 1F00A   7336    WAIT101:  JSR WAIT103$                    ;RESTORE R14 & R15 AND...
01001 FFFF E0E83   7337              JMP ISSCMDA2                    ;TRY ISSUING COMMAND AGAIN
                   7338    ;
                   7339    WAIT102:  MOVSMB SAVCYL,NEXTCYL           ;Restore saved Next Cylinder
01002 373F0 F01FE  7339 +            RDRAMB   R0,SAVCYL
01003 3410F F0026  7339 +            ALDRAM   R0,NEXTCYL
                   7340              MOVSMB SAVHD,NEXTHD             ;Restore saved Next Head
01004 373F0 F01FF  7340 +            RDRAMB   R0,SAVHD
01005 3410F F0027  7340 +            ALDRAM   R0,NEXTHD
                   7341              MOVSMB SAVREC,NEXTREC           ;Restore saved Next Record
01006 373F0 F023A  7341 +            RDRAMB   R0,SAVREC
01007 3410F F0028  7341 +            ALDRAM   R0,NEXTREC
                   7342              MOVSMB SAV415,SEN415            ;Restore saved Retry count
01008 373F0 F023F  7342 +            RDRAMB   R0,SAV415
01009 3410F F0018  7342 +            ALDRAM   R0,SEN415
                   7343
                   7344    WAIT103:  LDWD R15,R15SAVU                ;Restore saved R15 flag reg.
0100A 373FF F01F3  7344 +            RDRAMB   R15,R15SAVU              ;GET 12 MSBS
0100B D0FFF FFFFF  7344 +            RDXFR    SMC,,R15                 ;PUT IN UPPER REGISTER
0100C 353FF F01F4  7344 +            OR DB,B,RAME,R15 & NOP R15SAVU+1  ;OR IN 12 LSBS
0100D 373FE F01F2  7345              RDRAMB R14,R14SAVL              ;RESTORE ALT. FLAGS
0100E 373FD AF0B4  7346              RDREGB RAME,R13 & RTS ILDDRV    ;RESTORE ILD DRIVE & RETURN
                   7347
                   7348    WAIT100:  MOVSMB NEXTCYL,SAVCYL           ;Save Next Cyl. thru Wait Loop
0100F 373F0 F0026  7348 +            RDRAMB   R0,NEXTCYL
01010 3410F F01FE  7348 +            ALDRAM   R0,SAVCYL
                   7349              MOVSMB NEXTHD,SAVHD             ;Save Next Head thru Wait Loop
01011 373F0 F0027  7349 +            RDRAMB   R0,NEXTHD
01012 3410F F01FF  7349 +            ALDRAM   R0,SAVHD
                   7350              MOVSMB NEXTREC,SAVREC           ;Save Next Rec. thru Wait Loop
01013 373F0 F0028  7350 +            RDRAMB   R0,NEXTREC
01014 3410F F023A  7350 +            ALDRAM   R0,SAVREC
                   7351              MOVSMB SEN415,SAV415            ;Save Retry count thru Wait Loop
01015 373F0 F0018  7351 +            RDRAMB   R0,SEN415
01016 3410F F023F  7351 +            ALDRAM   R0,SAV415
                   7352    WAIT100A: SVWD R15,R15SAVU                ;SAVE R15 thru the Wait Loop
01017 349FF F01F3  7352 +            ALDRAMU  R15,R15SAVU              ;SAVE 12 MSBS
01018 341FF F01F4  7352 +            ALDRAM   R15,R15SAVU+1            ;SAVE 12 LSBS
01019 341EF F01F2  7353              ALDRAM R14,R14SAVL              ;SAVE R14
                   7354              LDAR.P WAIT00                   ;WAIT LOOP AFTER DISPLAY ROUTINE
0101A E0400 5F002  7354 +            SETPGE WAIT00_SHR_H#B & LDARP WAIT00
```

```
 Addr            Line ILD   MISC. ROUTINES

                 7355    ;
0101B 373F3 F00BA 7356            RDRAMB R3,ILDFLG                ;GET ILD FLAGS
0101C 45053 F0040 7357            TSTRBIT R3,40                   ;CHECK BYPASS WAIT FLAG
0101D 433FF 3301F 7358            RESETR R15 & BOZ $+2            ;IF NOT, JUMP
                  7359            LDAR.P WAIT17                   ;DON'T GO THRU THE WAIT LOOP
0101E E0400 5F022 7359 +          SETPGE WAIT17_SHR_H#B & LDARP WAIT17
0101F 3710F F0242 7360            RESETRAM CASHSTAT               ;RESET CACHING STATUS
01020 433F9 E002D 7361            RESETR R9 & JMP DISP00          ;DISPLAY ROUTINE; RESET TIMER
                  7362    ;
                  7363    TITLE IN LINE DIAGNOSTICS
```

Addr Line IN LINE DIAGNOSTICS

```
7365
7366    ;    ILD FLAGS
7367    ;BIT    MEANING
7368    ;
7369    ;  0    ILD ACTIVE
7370    ;  1    ILD SEEK COMPLETE PENDING
7371    ;  2    ILD RUNNING
7372    ;__3___ILD SEEK INITIATED
7373    ;  4    FORMAT-TO-VERIFY FLAG
7374    ;  5    LAST OPERATION
7375    ;  6    BYPASS WAIT LOOP
7376    ;__7___(not used)
7377    ;  8    ALTERNATE CYL./REV. SEEK FLAG
7378    ;  9    DEFECTIVE ALTERNATE
7379    ; 10    FORMAT DEFECTIVE
7381    ;_11___$ (not used)
7385    ;
7386    ;MSGFLG:  EQU H#1EF       ;DISPLAY 4A5       ;MESSAGE SENT FLAGS / LOOP FLAG
7387    ;         LOOPFLG:  EQU H#1                  ;ILD Loop-on-error flag
7388    ;         MSG1:     EQU H#2                  ;ILD Message sent-1 flag
7389    ;         MSG2:     EQU H#4                  ;ILD Message sent-2 flag
7393
```

Addr Line IN LINE DIAGNOSTICS

```
                    7395
01021 0C00F C0003   7396    ILD00:    ADD AZC,Q,R0 & LDCT 3              ;ILD NO.+1 TO Q
01022 0E12E 5F034   7397              ADD DQC,FOUTL,SEQAR,IREG & LDAR ILD02 ;JUMP LOC. FOR VALIDITY CHECK
01023 37929 FFFFF   7398              OR DZ,FOUTU,SEQAR,IREGH
01024 0F10F 6F08A   7399              ADD DZC,FOUTL,ZERO,RAME & JIR ILDFLG ;SET ILD ACTIVE FLAG
                    7400    ;
01025 FFFFF 5FFFF   7401              JAR                                ;ILD-1  FORMAT THE PACK
01026 FFFFF 5FFFF   7402              JAR                                ;ILD-2  VERIFY THE PACK
01027 FFFFF 5FFFF   7403              JAR                                ;ILD-3  VERIFY ONE TRACK
01028 FFFFF 5FFFF   7404              JAR                                ;ILD-4  FORMAT DEFECTIVE TRACK
01029 FFFFF 5FFFF   7405              JAR                                ;ILD-5  OSCILLATING SEEK
0102A FFFFF 5FFFF   7406              JAR                                ;ILD-6  RANDOM SEEK TEST
0102B FFFFF 5FFFF   7407              JAR                                ;ILD-7  SEQUENTIAL SEEK TEST
0102C FFFFF 5FFFF   7408              JAR                                ;ILD-8  DISK ECC DISPLAY CLEAR
0102D FFFFF E10A8   7410              JMP      ILD08                     ;$ILD-9 Invalid ILD for CDC
0102E FFFFF E10A8   7411              JMP      ILD08                     ;$ILD-A Invalid ILD for CDC
0102F FFFFF 5FFFF   7412              JAR                                ;$ILD-B Special CDC ILD
01030 FFFFF E10A8   7418              JMP ILD08                          ;ILD-C  Bad Entry
01031 FFFFF E10A8   7419              JMP ILD08                          ;ILD-D  BAD ENTRY
01032 FFFFF E10A8   7420              JMP ILD08                          ;ILD-E  BAD ENTRY
01033 FFFFF E10A8   7421              JMP ILD08                          ;ILD-F  BAD ENTRY
                    7422    ;
                    7423    ;   Print "ILD n" where n = ILD number.
                    7424    ;
01034 3410F F00BD   7425    ILD02:    ALDRAM R0,ILDNBR                   ;SAVE ILD NUMBER
01035 01700 98035   7426              ADD AB,LSBZ,R0,R0 & CJPCNT EVCNTZ,$ ;MOVE ILD NO. INTO POSITION AND..
01036 3410F F00B1   7427              ALDRAM R0,MAINILD1                 ;DISPLAY IT
01037 37350 F0037   7428              RDMSKB R0,D#55
01038 0F322 E0E7B   7429              CALL R2,ISSCMDA)                   ;MSG: "ILD "
01039 373F0 F00BD   7430              RDRAMB R0,ILDNBR                   ;GET ILD NUMBER
0103A 37351 F0001   7431              RDMSKB R1,1                        ;1 CHAR.
0103B 3705F F0008   7432              RDMSKQ 8
0103C 0F322 E0E7C   7433              CALL R2,ISSCMDA                    ;PRINT ILD NUMBER
                    7434    ;
0103D 3710F F01F2   7435              RESETRAN R14SAVL                   ;RESET R14 SAVE LOC.
0103E 373F0 F00BD   7436              RDRAMB R0,ILDNBR                   ;GET ILD NUMBER
0103F 1D040 F0008   7437              SUBA DB,Q,MSK1,R0 & NOP H#8        ;CHECK FOR A DRIVE ILD
01040 433FE 35049   7438              RESETR R14 & CJMP ALB,ILD03        ;IF DRIVE ILD, JUMP; RESET FLAGS
01041 0615E F1044   7439              ADD DQ,FOUTL,MSK2,IREG & NOP ILD02A ;COMPUTE NON-DRIVE JUMP LOC.
01042 37929 FFFFF   7440              OR DZ,FOUTU,SEQAR,IREGH
01043 3710F 6F1EF   7441              MOVE ZERO,RAME & JIR MSGFLG        ;RESET LOOP & MSG. SENT FLAGS
                    7442    ;
01044 FFFFF E1582   7443    ILD02A:   JMP ILD800                         ;CLEAR DISK ECC DISPLAY
01045 FFFFF FFFFF   7445              NOP                                ;$
01046 FFFFF FFFFF   7446              NOP                                ;$
01047 FFFFF E158E   7447              JMP      CDCILD                    ;$Go to special CDC ILD
01048 FFFFF FFFFF   7453              NOP
                    7454    ;
```

Addr Line IN LINE DIAGNOSTICS

```
01049 37350 F10D2  7455    ILD03:    RDMSKB R0,ILD20
                   7456              JSR.P SAVSMRTN                    ;SAVE LOOP-ON-ERROR RETURN
0104A E0400 1F4C6  7456 +            SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
                   7457    ;
0104B 3710F F01EF  7458              RESETRAM MSGFLG                   ;RESET MESSAGE SENT FLAGS
0104C 373FD F00B4  7459              RDRAMB R13,ILDDRV                 ;GET LAST DRIVE NO.
0104D 0F300 33065  7460              ADD DZC,B,ZERO,R0 & BOZ ILD04     ;IF NONE, JUMP
0104E 5535D F0800  7461              RSTRBIT R13,VALID                 ;RESET VALID BIT
0104F 370FF F00B1  7465              RDRANQ MAINILD1                   ;GET ILD NO. DISPLAY AND...
01050 301DF F00B1  7466              OR AQ,FOUTL,R13,RAME & NOP MAINILD1 ;INCLUDE DRIVE NO.
                   7467    ;
                   7468    ;   Print the last ILD Drive number
                   7469    ;
01051 433F0 FFFFF  7470              RESETR R0
01052 0F322 E0E7B  7471              CALL R2,ISSCMDA1                  ;MSG: ", DR.="
01053 0F322 E126E  7472              CALL R2,PRNTDRV                   ;PRINT DRIVE NO.
                   7473    ;
01054 37352 F10B2  7474              RDMSKB R2,ILD10
01055 FFFFF 1F2F9  7475              JSR SAVRET5                       ;SAVE SELECTION ERROR RETURN
01056 433FC 1F2FB  7476              RESETR R12 & JSR SELDRIV          ;SELECT ILD DRIVE
                   7477    ;
01057 0F322 E125B  7478              CALL R2,YESNO                     ;CHECK IF DRIVE NO. IS O.K.
01058 433F0 33065  7479              RESETR R0 & BOZ ILD04             ;IF NOT, GET NEW DRIVE NO.
                   7481              JSR.P    CLRFLAG                  ;$Clear flags this drive
01059 E0401 1F344  7481 +            SETPGE CLRFLAG_SHR_H#B & JSRP CLRFLAG
0105A 270FF F00BD  7485              SAB DZ,Q,RAME & NOP ILDNBR        ;GET ILD NO.-1
0105B 0615E F105E  7486              ADD DQ,FOUTL,MSK2,IREG & NOP ILD03A ;COMPUTE JUMP LOC.
0105C 37959 F105E  7487              OR DZ,FOUTU,MSK2,IREGH & NOP ILD03A
0105D 433FB 6F7FF  7488              RESETR R11 & JIR
                   7489    ;
0105E FFFFF E1309  7490    ILD03A:   JMP ILD100                        ;DISPLAY LAST FORMAT PARAMETERS
0105F FFFFF E1309  7491              JMP ILD100                        ;DISPLAY LAST VERIFY PARAMETERS
01060 FFFFF E148B  7492              JMP ILD300                        ;DISPLAY READ VERIFY CYL. & HEAD
01061 FFFFF E1C9C  7493              JMP ILD400+1                      ;GET DEFECTIVE CYL. & HEAD
01062 FFFFF E14B0  7494              JMP ILD500                        ;DISPLAY SEEK CYLINDERS
01063 FFFFF E14E5  7495              JMP ILD600                        ;GO TO RANDOM SEEK TEST
01064 FFFFF E14B0  7496              JMP ILD500                        ;DISPLAY START & ENDING CYLS.
                   7504    ;
                   7505    ;    GET NEW DRIVE NUMBER
                   7506    ;
01065 3710F F00B5  7507    ILD04:    RESETRAM CURCYL                   ;RESET CURRENT CYLINDER,
01066 3710F F00B6  7508              RESETRAM CURHD                    ;CURRENT HEAD,
01067 3710F F00BB  7509              RESETRAM STHD                     ;STARTING HEAD AND...
01068 3710F F00BC  7510              RESETRAM NDHD                     ;ENDING HEAD
01069 433FE FFFFF  7511              RESETR R14
                   7512    ;
```

IN LINE DIAGNOSTICS

```
Addr            Line

0106A 373F1 F00B1  7513   ILD05:   RDRAMB R1,MAINILD1               ;GET DRIVE NO. DISPLAY
0106B 55351 F005F  7514            RSTRBIT R1,5F                   ;DELETE PREVIOUS DRIVE NO. AND...
0106C 3411F F00B1  7515            ALDRAM R1,MAINILD1              ;DISPLAY ONLY ILD NO.
                   7516   ;
0106D 3710F F00B2  7517            RESETRAM MAINILD2               ;RESET THE 24 LSB'S OF...
0106E 3710F F00B3  7518            RESETRAM MAINILD3               ;THE MAIN ILD DISPLAY
0106F 3710F F00B4  7519            RESETRAM ILDDRV                 ;RESET DRIVE NUMBER
01070 05340 F0005  7520            ADD DB,B,MSK1,R0 & NOP D#5
01071 0F322 E0E7B  7521            CALL R2,ISSCMDA1                ;MSG: "ENTER DRIVE NO. "
01072 3705F F0087  7522            RDMSKQ 87
01073 37350 F0002  7523            RDMSKB R0,2                     ;2 DIGITS
01074 0F322 E0E7C  7524            CALL R2,ISSCMDA                 ;GET NEW DRIVE NUMBER
                   7525   ;
01075 2D040 F001F  7526            SUBB DB,Q,MSK1,R0 & NOP D#31    ;CHECK FOR VALID DRIVE NO.
01076 FFFFF 150B0  7527            CJSR ALB,ILD09                  ;IF NOT, JUMP
01077 3430D FFFFF  7528            RTOR R0,R13                     ;DRIVE NO. TO R13
01078 340DF FFFFF  7532            RTOQ R13                        ;UNIT NO. TO Q
01079 373F1 F00B1  7533            RDRAMB R1,MAINILD1              ;GET DISPLAYED ILD NO. AND...
0107A 3011F F00B1  7534            OR AQ,FOUTL,R1,RAME & NOP MAINILD1 ;MERGE NEW DRIVE NO.
0107B 3535D F0800  7535            SETRBIT R13,VALID               ;ADD VALID BIT TO DRIVE NO.
                   7537            JSR.P    CLRFLAG                ;$Clear flags this drive
0107C E0401 1F344  7537 +          SETPGE CLRFLAG_SHR_H#B & JSRP CLRFLAG
0107D 433FB FFFFF  7543            RESETR R11                      ;RESET CYLINDER REGISTER
0107E 341DF F00B4  7544            ALDRAM R13,ILDDRV               ;SAVE DRIVE NO.
                   7545   ;
0107F 37352 F10B1  7546            RDMSKB R2,$+2                   ;SAVE CONTINUATION LOCATION...
01080 FFFFF 1F183  7547            JSR SAVCONT                     ;IN CASE OF ERROR
01081 37352 F0AF3  7548            RDMSKB R2,SEL22
01082 FFFFF 1F2F9  7549            JSR SAVRET5                     ;SAVE SELECTION ERROR JUMP LOC.
01083 433FC 1F2FB  7550            RESETR R12 & JSR SELDRIV        ;SELECT ILD DRIVE
01084 3710F F01EF  7551            RESETRAM MSGFLG                 ;DRIVE SELECTED, RESET MSG. FLAGS
                   7552   ;
01085 370FF F00BD  7553            RDRAMQ ILDNBR                   ;GET ILD NUMBER
01086 0615E F1088  7554            ADD DQ,FOUTL,MSK2,IREG & NOP $+D#2 ;COMPUTE JUMP LOC.
01087 37929 FFFFF  7555            OR DZ,FOUTU,SEQAR,IREGH
01088 373FD 6F0B4  7556            RDREGB RAME,R13 & JIR ILDDRV    ;GET ILD DRIVE NO. & JUMP
                   7557   ;
01089 FFFFF E1346  7558            JMP ILD112                      ;DISPLAY PARAM. LIMITS
0108A FFFFF E1346  7559            JMP ILD112                      ;DISPLAY PARAM. LIMITS
0108B 433F0 E1499  7560            RESETR R0 & JMP ILD304          ;GET CYLINDER & HEAD ENTRIES
0108C FFFFF E1C9B  7561            JMP ILD400                      ;GET DEFECTIVE CYL. & HEAD
0108D FFFFF E14B0  7562            JMP ILD500                      ;DISPLAY OR GET CYLINDERS
0108E FFFFF E14E5  7563            JMP ILD600                      ;GO TO RANDOM SEEK TEST
0108F FFFFF E1514  7564            JMP ILD700                      ;DISPLAY CYLINDER LIMITS
                   7565   ;
01090 0F322 E125B  7566   ILD06:   CALL R2,YESNO                   ;ARE THESE PARAMS. OK?
01091 FFFFF 3309D  7567            BOZ ILD07                       ;IF NOT, JUMP
01092 370FF F00BD  7568   ILD06A:  RDRAMQ ILDNBR                   ;GET ILD NUMBER
01093 0615E F1095  7569            ADD DQ,FOUTL,MSK2,IREG & NOP $+D#2 ;COMPUTE JUMP LOC.
```

Addr Line IN LINE DIAGNOSTICS

```
01094 37929 FFFFF  7570            OR DZ,FOUTU,SEQAR,IREGH
01095 FFFFF 6F7FF  7571            JIR
                   7572    ;
01096 FFFFF E1353  7573            JMP ILD115                      ;ASK THE FORMAT QUESTION
01097 FFFFF E1439  7574            JMP ILD200                      ;START VERIFY
01098 FFFFF E14A0  7575            JMP ILD306                      ;START ONE TRACK VERIFY
01099 FFFFF E1133  7576            JMP ILD28
0109A FFFFF E14CA  7577            JMP ILD506                      ;START OSCILLATING SEEK
0109B FFFFF E1133  7578            JMP ILD28                       ;IMPOSSIBLE
0109C FFFFF E151B  7579            JMP ILD702                      ;START SEQUENTIAL SEEK
                   7580    ;
                   7581    ;   Get new parameters
                   7582    ;
0109D 370FF F00BD  7583    ILD07:  RDRAMQ ILDNBR                   ;GET ILD NUMBER
0109E 0615E F10A0  7584            ADD DQ,FOUTL,MSK2,IREG & NOP $+D#2 ;COMPUTE JUMP LOC.
0109F 37929 FFFFF  7585            OR DZ,FOUTU,SEQAR,IREGH
010A0 FFFFF 6F7FF  7586            JIR
                   7587    ;
010A1 FFFFF E1338  7588            JMP ILD110                      ;GET NEW PARAMS.
010A2 FFFFF E1338  7589            JMP ILD110                      ;GET NEW PARAMS.
010A3 433F0 E1499  7590            RESETR R0 & JMP ILD304          ;GET NEW CYL. & HEAD
010A4 FFFFF E1133  7591            JMP ILD28                       ;TERMINATE THE ILD
010A5 433F0 E14C1  7592            RESETR R0 & JMP ILD503          ;GET NEW CYLINDERS
010A6 FFFFF E1133  7593            JMP ILD28                       ;IMPOSSIBLE
010A7 433F0 E14C1  7594            RESETR R0 & JMP ILD503          ;GET NEW CYLINDERS
                   7608
                   7609    ;   OPERATOR ENTERED A NUMBER OF A NON-EXISTENT OR ILLEGAL ILD.
                   7610    ;
010A8 0F322 E11F9  7611    ILD08:  CALL R2,BADNTRY                 ;MSG: "BAD ENTRY. "
010A9 3710F F00B1  7612            RESETRAM MAINILD1               ;RESET...
010AA 3710F F00B2  7613            RESETRAM MAINILD2               ;MAIN ILD...
010AB 3710F F00B3  7614            RESETRAM MAINILD3               ;DISPLAY
010AC 37350 F0038  7615            RDMSKB R0,D#56
010AD 0F322 E0E7B  7616            CALL R2,ISSCMDA1                ;MSG: "ENTER ILD NO. "
010AE 3710F F00BA  7617            RESETRAM ILDFLG                 ;RESET ILD FLAGS AND...
010AF FFFFF E0002  7618            JMP WAIT00                      ;GO TO THE WAIT LOOP
                   7619
                   7620
                   7621    ;   OPERATOR ENTERED AN ILLEGAL DRIVE NUMBER.
                   7622    ;
010B0 0F322 E11F9  7623    ILD09:  CALL R2,BADNTRY                 ;MSG: "BAD ENTRY. "
010B1 0F300 BF06A  7624            ADD DZC,B,ZERO,R0 & JPOP ILD05  ;GET NEW DRIVE NO.
                   7625
                   7626
                   7627    ;   INITIAL DRIVE SELECTION WAS UNSUCCESSFUL.  THIS ROUTINE WILL PRINT THE
                   7628    ;MESSAGE "DRIVE NOT READY", DELAY FOR A WHILE, AND THEN REQUEST A NEW DRIVE
                   7629    ;NUMBER.
                   7630    ;
010B2 37350 F0026  7631    ILD10:  RDMSKB R0,D#38
010B3 0F322 E0E7B  7632            CALL R2,ISSCMDA1                ;MSG: "DRIVE NOT READY"
010B4 37350 F3800  7633            RDMSKB R0,3800                  ;14,336 passes thru Wait Loop
```

Addr Line IN LINE DIAGNOSTICS

```
010B5 37352 F10BA  7641    ILD10A:   RDMSKB R2,ILD10B                ;RETURN LOC. FROM WAIT LOOP
010B6 3490F F01FB  7642              ALDRAMU R0,PASCNT1
010B7 3410F F01FC  7643              ALDRAM R0,PASCNT
                   7644              JSR.P SAVRET                    ;SAVE RETURN LOC. AND...
010B8 E0401 1F69B  7644 +            SETPGE SAVRET_SHR_H#B & JSRP SAVRET
010B9 FFFFF E100F  7645              JMP WAIT100                     ;GO THRU THE WAIT LOOP
                   7646    ;
010BA 373F1 F01FC  7647    ILD10B:   RDRAMB R1,PASCNT                ;GET THE PASS COUNT
010BB 133F1 1308E  7648              DECRB R1 & CJSR ALUZ,ILD11      ;DECR. IT; IF 0, JUMP
010BC 3411F F01FC  7649              ALDRAM R1,PASCNT                ;SAVE NEW COUNT AND...
010BD FFFFF E1012  7650              JMP WAIT100+3                   ;RETURN TO THE WAIT LOOP
                   7651    ;
010BE 373F0 F01FB  7652    ILD11:    RDRAMB R0,PASCNT1               ;GET PASS COUNT MSB'S
010BF FFFFF B30C1  7653              CJPUP ALUZ,$+2                  ;IF 0, JUMP
010C0 1410F AF1FB  7654              SBA AZ,FOUTL,R0,RAME & RTS PASCNT1 ;SAVE NEW MSB'S -1 & RETURN
010C1 433FF E1065  7655              RESETR R15 & JMP ILD04          ;GET NEW DRIVE NO. & RESET FLAGS
                   7656
                   7657
                   7658    TITLE ILD ENDING ROUTINE
                   7659
                   7660
                   7661    ;    THE ILD ENDING ROUTINE IS ENTERED FROM END51 IF THE ILD RUNNING FLAG IS
                   7662    ;SET.  THE ROUTINE HANDLES ERROR DISPLAYS & MESSAGES, LOOPING ON ERRORS AND
                   7663    ;OPERATOR INTERFACING.
                   7664    ;
010C2 373FE F01F2  7665    ILDEND:   RDRAMB R14,R14SAVL              ;RESTORE SAVED R14 FLAGS
010C3 4505D F0200  7676              TSTRBIT R13,200                 ;TEST FOR UNIT CHECK
010C4 FFFFF 338F0  7677              BNZ ILD25                       ;IF UNIT CHECK, JUMP
                   7678    ;
010C5 6505A F0010  7679              TSTRMSK R10,10                  ;CHECK STATUS ACTION = 4
010C6 FFFFF 330DF  7680              BOZ ILD23                       ;IF YES, SEEK IN PROGRESS
010C7 6505A F0014  7681              TSTRMSK R10,14                  ;CHECK STATUS ACTION = 5
010C8 FFFFF 330E3  7682              BOZ ILD24                       ;IF YES, SEEK COMPLETE
                   7683    ;
010C9 45057 F0020  7684              TSTRBIT R7,20                   ;CHECK LAST OPERATION FLAG
010CA FFFFF 3394E  7685              BNZ ILD32                       ;IF SET, VERIFY IS COMPLETE
                   7686              JSR.P CHKDATA                   ;CHECK FORMAT DATA
010CB E0403 1F259  7686 +            SETPGE CHKDATA_SHR_H#B & JSRP CHKDATA
                   7687    ;
                   7688    ILD19:    EQU $
010CC 373F1 F01EF  7694              RDRAMB R1,MSGFLG                ;GET MSG. SENT & LOOP FLAGS
010CD 45351 F0007  7695              MSKRBIT R1,MSG2+MSG1+LOOPFLG    ;Isolate the flags
010CE 65051 F0007  7696              TSTRMSK R1,MSG2+MSG1+LOOPFLG    ;CHECK FOR ALL FLAGS SET
010CF 0F301 338D2  7697              ADD DZC,B,ZERO,R1 & BNZ ILD20   ;IF NOT, CONTINUE
010D0 3411F F01EF  7698              ALDRAM R1,MSGFLG                ;Set only loop-on-error flag
010D1 FFFFF 1F233  7699              JSR ILDDSP                      ;DISPLAY ILD & DRIVE NUMBERS
                   7700    ;
                   7701    ;   Go thru the Wait Loop
                   7702    ;
```

```
 Addr            Line ILD ENDING ROUTINE

010D2 FFFFF 1F280  7703   ILD20:   JSR ILDCYHD                    ;SAVE CUR. CYL. & HEAD
010D3 37352 F10D6  7704            RDMSKB R2,ILD21                ;RETURN LOC. FROM THE WAIT LOOP
                   7705            JSR.P SAVRET                   ;SAVE IT AND...
010D4 E0401 1F69B  7705 +          SETPGE SAVRET_SHR_H#B & JSRP SAVRET
010D5 FFFFF E100F  7706            JMP WAIT100                    ;GO THROUGH THE WAIT LOOP
                   7707   ;
                   7708   ;   Return from the Wait Loop
                   7709   ;
010D6 FFFFF 1F002  7710   ILD21:   JSR WAIT102                    ;RESTORE THE REGISTERS
010D7 0F322 E1217  7711            CALL R2,ILDHALT                ;CHECK FOR ILD OPERATOR HALT
010D8 0F322 E1239  7712   ILD22:   CALL R2,ILDRUN                 ;RE-SELECT DRIVE IF NECESSARY
010D9 5535F F8108  7713            RSTRBIT R15,8108               ;RESET SELECTED FLAGS
010DA 37350 F10D2  7714            RDMSKB R0,ILD20                ;SAVE SM-ERROR RETURN LOCATION...
                   7715            JSR.P SAVSMRTN                 ;IN CASE OF ERROR
010DB E0400 1F4C6  7715 +          SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
010DC 371FE F01F1  7716            MOVE RAME,IREG & NOP ILDCONT   ;RESTORE THE CONTINUATION...
010DD 371F9 F01F0  7717            MOVE RAME,IREGH & NOP ILDCONTA ;RETURN LOCATION AND...
010DE FFFFF 6F7FF  7719            JIR                            ;$Continue the ILD
                   7723   ;
                   7724   ;   SEEK IN PROGRESS
                   7725   ;
010DF 55357 F000C  7726   ILD23:   RSTRBIT R7,H#C                 ;RESET SEEK INIT./ILD RUN FLAGS
010E0 3417F F00BA  7727            ALDRAM R7,ILDFLG
                   7728            JSR.P RELDRIV                  ;RELEASE & DE-SELECT THE DRIVE
010E1 E0401 1F60F  7728 +          SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
010E2 FFFFF E100F  7729            JMP WAIT100                    ;GO TO THE WAIT LOOP
                   7730   ;
                   7731   ;    SEEK COMPLETE
                   7732   ;
010E3 55357 F000A  7733   ILD24:   RSTRBIT R7,H#A                 ;RESET SEEK INIT/SEEK COMP. FLAGS
010E4 3417F F00BA  7734            ALDRAM R7,ILDFLG
010E5 433F0 FFFFF  7735            RESETR R0
010E6 3715E F0063  7736            MOVE MSK2,IREG & NOP SKILD     ;CLEAR ILD SEEK COMPLETE...
                   7737            JSR.P SMBWRT                   ;PENDING BIT FOR THIS DRIVE
010E7 E0400 1F044  7737 +          SETPGE SMBWRT_SHR_H#B & JSRP SMBWRT
                   7738            JSR.P WAIT102                  ;RESTORE REGISTERS AND...
010E8 E0402 1F002  7738 +          SETPGE WAIT102_SHR_H#B & JSRP WAIT102
010E9 373F1 F01EF  7739            RDRANB R1,MSGFLG               ;GET LOOP & MSG. SENT FLAGS
010EA 45351 F0007  7740            MSKRBIT R1,MSG2+MSG1+LOOPFLG   ;Isolate the flags
010EB 65051 F0007  7741            TSTRMSK R1,MSG2+MSG1+LOOPFLG   ;ARE THEY ALL SET?
010EC E100B 33988  7742            CTL MSSM,RSTUSEL & BNZ RTRN3   ;IF NOT, RETURN TO THE ILD
010ED 0F10F F01EF  7743            ADD DZC,FOUTL,ZERO,RAME & NOP MSGFLG ;RESTORE LOOP FLAG ONLY
010EE FFFFF 1F233  7744            JSR ILDDSP                     ;DISPLAY ILD & DRIVE NO.
010EF FFFFF E1188  7745            JMP RTRN3                      ;RETURN TO THE ILD
```

ILD ENDING ROUTINE

```
Addr            Line
                7747    ;
                7748    ;   UNIT CHECK STATUS
                7749    ;
010F0 55357 F000A 7750  ILD25:    RSTRBIT R7,H#A                ;RESET SEEK INIT/SEEK COMP. FLAGS
010F1 3417F F00BA 7751            ALDRAM R7,ILDFLG
010F2 5535D F0080 7752            RSTRBIT R13,80                ;RESET S.S.S. UNIT FLAG
010F3 341AF F00B1 7753            ALDRAM R10,MAINILD1           ;DISPLAY ERROR...
010F4 349DF F00B2 7754            ALDRAMU R13,MAINILD2          ;STATUS IN...
010F5 341DF F00B3 7755            ALDRAM R13,MAINILD3           ;THE LIGHTS
010F6 3415F F023E 7756            ALDRAM R5,R5SAVL              ;IN CASE OF WMA ERROR...
010F7 3495F F023D 7757            ALDRAMU R5,R5SAVU             ;SAVE R5 FOR RETRY
                7758    ;
                7759    ;   ROUTINE TO SAVE ILD SENSE BYTES  (DISPLAY 4B0-4BC)
                7760    ;
010F8 3705F F0210 7761  ILD25B:   RDMSKQ 210                    ;OFFSET BETWEEN SENSE & ILD SENSE
010F9 3710E 4F025 7762            MOVE ZERO,IREG & PSHLD H#25   ;LOC. OF SENSE BYTE 0
010FA 373D0 FFFFF 7763            RDREGB RAM1,R0                ;GET THE SENSE BYTE
010FB 061EE FFFFF 7764            ADD DQ,FOUTL,IREG,IREG        ;ADD OFFSET AND...
010FC 3410D FFFFF 7765            ALDREG R0,RAM1                ;SAVE ILD SENSE BYTE
010FD 2E1EE FFFFF 7766            SUBB DQ,FOUTL,IREG,IREG       ;COMPUTE LOC. OF NEXT SENSE BYTE
010FE 0F1EE 887FF 7767            INCRD IREG,IREG & CLPCNT EVCNTZ ;IF NOT DONE, CONTINUE
                7768    ;
010FF 3710F F0018 7769            RESETRAM SEN415               ;RESET SM-RETRY COUNT
                7770              TSTMBIT LOOPFLG,MSGFLG        ;Loop-on-error flag set?
01100 373F0 F01EF 7770 +          RDRAMB   R0,MSGFLG
01101 45150 F0001 7770 +          TSTBIT   R0,LOOPFLG
01102 FFFFF 33945 7771            BNZ ILD31                     ;YES, LOOP ON ERROR
                7772    ;
                7773              TSTMEQU  6,SEN03              ;Recovery Action
01103 373F0 F0003 7773 +          RDRAMB   R0,SEN03
01104 65150 F0006 7773 +          TSTREQU  R0,6
01105 65050 F0006 7774            TSTRMSK R0,6                  ;CHECK FOR RECOV. ACTION = 6
01106 FFFFF 33140 7775            BOZ ILD30                     ;IF 6, DRIVE NOT READY
```

Addr Line ILD ENDING ROUTINE

```
                    7777    ;
                    7778    ;   PRINT ERROR MESSAGE
                    7779    ;
01107 370FF F01F6   7780            RDRAMQ PARAM1                   ;GET LAST MESSAGE NO.
01108 6605F F0047   7781            TSTQMSK D#71                    ;CACHE SELECT ERROR?
01109 FFFFF 33123   7782            BUZ ILD27                       ;YES, MESSAGE HAS BEEN PRINTED
0110A 0F0FF F00BD   7783            ADD DZC,0,RAME & NOP ILDNBR     ;GET THE ILD NUMBER+1
0110B 0E12E 5F11D   7784            ADD DQC,FOUTL,SEQAR,IREG & LDAR ILD26 ;COMPUTE JUMP LOC.
0110C 37929 FFFFF   7785            OR DZ,FOUTU,SEQAR,IREGH
0110D FFFFF 6F7FF   7786            JIR                             ;GET ERROR MESSAGE
                    7787    ;
0110E 37340 5F817   7788            RDREGB MSK1,R0 & JAR D#23       ;(1) FORMAT ERROR
0110F 37340 5F818   7789            RDREGB MSK1,R0 & JAR D#24       ;(2) VERIFY ERROR
01110 37340 5F818   7790            RDREGB MSK1,R0 & JAR D#24       ;(3) VERIFY ERROR
01111 37340 5F817   7791            RDREGB MSK1,R0 & JAR D#23       ;(4) FORMAT ERROR
01112 37340 5F839   7792            RDREGB MSK1,R0 & JAR D#57       ;(5) SEEK ERROR
01113 37340 5F839   7793            RDREGB MSK1,R0 & JAR D#57       ;(6) SEEK ERROR
01114 37340 5F839   7794            RDREGB MSK1,R0 & JAR D#57       ;(7) SEEK ERROR
01115 37340 5F840   7795            RDREGB MSK1,R0 & JAR D#64       ;(8) CLEAR ECC DISP. ERROR
01116 FFFFF FFFFF   7797            NOP                             ;$(9) (not used)
01117 FFFFF FFFFF   7798            NOP                             ;$(A) (not used)
01118 FFFFF FFFFF   7803            NOP                             ;(B)
01119 FFFFF FFFFF   7804            NOP                             ;(C)
0111A FFFFF FFFFF   7805            NOP                             ;(D)
0111B FFFFF FFFFF   7806            NOP                             ;(E)
0111C FFFFF FFFFF   7808            NOP                             ;$(F) not used
                    7812    ;
0111D 0F322 E0E7B   7813    ILD26:  CALL R2,ISSCMDA1                ;PRINT ERROR MESSAGE
0111E 373F1 F00BD   7814            RDRAMB R1,ILDNBR                ;GET ILD NUMBER
0111F 65051 F0008   7815            TSTRMSK R1,8                    ;CHECK FOR ILD-8
01120 FFFFF 33123   7816            BUZ ILD27                       ;IF YES, SKIP DRIVE INFO.
                    7823    ;
                    7824    ;   Print Drive error info.
                    7825    ;
01121 0F322 E126E   7826            CALL R2,PRNTDRV                 ;PRINT DRIVE NUMBER
01122 0F322 E1289   7827            CALL R2,CYLHED                  ;PRINT CYL. & HEAD NOS.
                    7828    ;
01123 3705F F000B   7829    ILD27:  RDMSKQ H#B
01124 0F322 E0E7C   7830            CALL R2,ISSCMDA                 ;STORE CURRENT MESSAGE
01125 37350 F0022   7831            RDMSKB R0,D#34                  ;MSG: "ERROR: RETRY=1, LOOP=2"
01126 3705F F008E   7832            RDMSKQ H#8E                     ;ALTERNATE MESSAGES
01127 0F322 E0E7C   7833            CALL R2,ISSCMDA                 ;WAIT FOR OPERATOR RESPONSE
                    7834    ;
01128 3710F F01EF   7835    ILD27B: RESETRAM MSGFLG                 ;RESET MSG. SENT & LOOP FLAGS
01129 5535F F9908   7836            RSTRBIT R15,9908                ;RESET SELECTED FLAGS
0112A 5535E F0200   7837            RSTRBIT R14,200                 ;RESET ALT. TRACK FLAG
```

```
 Addr            Line ILD ENDING ROUTINE

0112B 3535F F0400  7838            SETRBIT R15,400                  ;SET SM-RETRY FLAG
0112C 65050 F0001  7839            TSTRMSK R0,1                     ;CHECK FOR RETRY
0112D FFFFF 33131  7840            BOZ ILD27C                       ;IF YES, TRY AGAIN
0112E 65050 F0002  7841            TSTRMSK R0,2                     ;CHECK FOR LOOP
0112F FFFFF 33933  7842            BNZ ILD28                        ;IF NOT, TERMINATE THE ILD
                   7843    ;
01130 0F10F F01EF  7844            ADD DZC,FOUTL,ZERO,RAME & NOP MSGFLG ;SET LOOP-ON-ERROR FLAG
01131 FFFFF 1F233  7845    ILD27C: JSR ILDDSP                       ;DISPLAY ILD & DRIVE NO.
01132 FFFFF E10D8  7846            JMP ILD22
                   7847    ;
                   7848    ;   OPERATOR HAS TERMINATED THE ILD
                   7849    ;
01133 37350 F0023  7850    ILD28:  RDMSKB R0,D#35
01134 0F322 E0E7B  7851            CALL R2,ISSCMDA1                 ;MSG: "ILD TERMINATED"
                   7852    ;
                   7853    ILD29:  EQU $
                   7855            JSR.P   CLRFLAG                  ;$Clear drive flags
01135 E0401 1F344  7855 +          SETPGE CLRFLAG_SHR_H#B & JSRP CLRFLAG
01136 3715E F0063  7859            MOVE MSK2,1REG & NOP SKILD
01137 FFFFF C0002  7860            LDCT 2                           ;CLEAR ILD SEEK COMPLETE...
                   7861            JSR.P CLR25                      ;PENDING BIT FOR THIS DRIVE
01138 E0400 1F33D  7861 +          SETPGE CLR25_SHR_H#B & JSRP CLR25
01139 FFFFF 1F26B  7862            JSR RSTIBIT                      ;CLEAR BIT FOR THE ILD DRIVE
                   7875    ;
0113A FFFFF 1F205  7876    ILD29A: JSR ILDREL                       ;RELEASE & DE-SELECT ILD DRIVE
0113B 3710F F00BD  7877            RESETRAM ILDNBR                  ;RESET ILD NUMBER
0113C 3710F F00BA  7878            RESETRAM ILDFLG                  ;RESET ILD FLAGS AND...
0113D 3710F F01EF  7879            RESETRAM MSGFLG                  ;MSG. SENT & LOOP FLAGS
0113E 3710F F0018  7880            RESETRAM SEN415                  ;RESET RETRY COUNT
0113F FFFFF E032C  7882            JMP     CLR15                    ;$Wait Loop via Clear routine
                   7886
                   7887    ;   DRIVE NOT READY
                   7888    ;
01140 3705F F000F  7889    ILD30:  RDMSKQ H#F                       ;COMMAND TO CLEAR MSG. DISPLAY
01141 0F322 E0E7C  7890            CALL R2,ISSCMDA                  ;CLEAR DISPLAY
01142 37350 F0026  7891            RDMSKB R0,D#38
01143 0F322 E0E7B  7892            CALL R2,ISSCMDA1                 ;MSG: "DRIVE NOT READY"
01144 FFFFF E1123  7893            JMP ILD27
                   7894    ;
                   7895    ;   LOOPING ON ERROR
                   7896    ;
01145 3535F F0400  7897    ILD31:  SETRBIT R15,400                  ;SET SM-RETRY FLAG
01146 5535E F0200  7898            RSTRBIT R14,200                  ;RESET ALT. TRACK FLAG
01147 45050 F0004  7899            TSTRBIT R0,MSG2                  ;CHECK FOR MSG. SENT 2 FLAG
01148 FFFFF 338D3  7900            BNZ ILD20+1                      ;IF SET, CONTINUE LOOPING
```

```
 Addr            Line ILD ENDING ROUTINE

01149 35350 F0006  7901            SETRBIT R0,MSG2+MSG1          ;SET MSG. SENT 1 & 2 FLAGS
0114A 3410F F01EF  7902            ALDRAM R0,MSGFLG
0114B 37350 F0032  7903            RDMSKB R0,D#50
0114C 0F322 E0E7B  7904            CALL R2,ISSCMDA1              ;MSG: "LOOPING ON ERROR"
0114D FFFFF E10D3  7905            JMP ILD20+1                   ;LOOP ON ERROR
                   7906   ;
                   7907   ;   VERIFY IS COMPLETE
                   7908   ;
                   7909   ILD32:   EQU $
0114E 341DF F00B3  7913            ALDRAM R13,MAINILD3           ;DISPLAY FINAL STATUS...
0114F 349DF F00B2  7914            ALDRAMU R13,MAINILD2          ;IN THE LIGHTS
01150 341AF F00B1  7915            ALDRAM R10,MAINILD1           ;(DISPLAY 43B)
01151 37350 F001A  7916            RDMSKB R0,D#26                ;MSG: "VERIFY COMPLETE"
                   7917   ;
01152 45057 F0200  7918            TSTRBIT R7,200                ;DEFECTIVE ALTERNATE FLAG?
01153 FFFFF 33155  7919            BOZ $+2                       ;IF NOT, JUMP
01154 37350 F0035  7920            RDMSKB R0,D#53                ;"DEFECTIVE ALTERNATE FORMAT COMPLETE"
                   7921   ;
01155 0F322 E0E7B  7922            CALL R2,ISSCMDA1              ;PRINT THE MESSAGE
                   7923            TSTMBIT  400,ILDFLG           ;Format Defective flag?
01156 373F0 F00BA  7923 +          RDRAMB   R0,ILDFLG
01157 45150 F0400  7923 +          TSTBIT   R0,400
01158 FFFFF 33135  7924            BOZ ILD29                     ;IF NOT, EXIT THE TEST
01159 FFFFF E1CF3  7925            JMP ILD411                    ;PRINT ALTERNATE TRACK NO.
                   7926
                   7927
                   7928   TITLE ILD SEEK ROUTINE
```

Addr Line ILD SEEK ROUTINE

```
                    7930
0115A FFFFF 1F181   7931    ILDSK0:   JSR SAVRET3                       ;SAVE SEEK RETURN
0115B 0F322 E1239   7932              CALL R2,ILDRUN                    ;SELECT ILD DRIVE IF NECESSARY
                    7933              MOVSMB   CURCYL,NEXTCYL           ;Set Next cyl.= Cur. cyl.
0115C 373F0 F00B5   7933 +            RDRAMB   R0,CURCYL
0115D 3410F F0026   7933 +            ALDRAM   R0,NEXTCYL
0115E 45050 F00FF   7934              TSTRBIT R0,H#FF                   ;ISOLATE 8 LSB'S IN Q AND...
0115F 321FF F0012   7935              QLDRAM SEN409                     ;SAVE THEM FOR SEEK
01160 60300 4F001   7936              XOR AQ,B,R0,R0 & PSHLD 1          ;ISOLATE MSB'S IN R0 AND...
                    7937              LSHFT2 R0                         ;Move them into position
01161 01700 887FF   7937 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01162 3490F F0011   7938              ALDRAMU R0,SEN408                 ;SAVE THEM FOR SEEK
                    7939              MOVSMB   CURHD,SEN411             ;Save head no. for Seek
01163 373F0 F00B6   7939 +            RDRAMB   R0,CURHD
01164 3410F F0014   7939 +            ALDRAM   R0,SEN411
                    7940    ;
01165 5535E F0200   7941              RSTRBIT R14,200                   ;RESET ALT. TRACK FLAG
01166 5535F F1800   7942              RSTRBIT R15,1800                  ;RESET ADV. CYL. & HEAD FLAGS
01167 35353 F0008   7947              SETRBIT R3,8                      ;SET SEEK INITIATE FLAG
01168 3413F F00BA   7948              ALDRAM R3,ILDFLG
                    7949    ;
                    7950              SVWD     R15,R15SAVU              ;Save the flag registers for...
01169 349FF F01F3   7950 +            ALDRAMU  R15,R15SAVU                ;SAVE 12 MSBS
0116A 341FF F01F4   7950 +            ALDRAM   R15,R15SAVU+1              ;SAVE 12 LSBS
0116B 341EF F01F2   7951              ALDRAM R14,R14SAVL                ;RETURN FROM SEEK
0116C 4505F F0008   7952              TSTRBIT R15,8                     ;CHECK RTZ FLAG
                    7953              BNZ.P SEEK20                      ;IF SET, RE-ZERO THE DRIVE
0116D E0400 33985   7953 +            SETPGE SEEK20_SHR_H#B & BNZP SEEK20
0116E 0F10F F0302   7954              ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag
0116F FFFFF E0127   7955              JMP SEEK00
                    7956    ;
                    7957    ;    THE SEEK OR RE-ZERO WILL RETURN TO THE ILD VIA THE ILDEND ROUTINE (ILD24).
                    7958
                    7959    TITLE ILD SUB-ROUTINES
```

Addr Line ILD SUB-ROUTINES

```
                    7961
                    7962    ;    THE FOLLOWING ROUTINE WILL DISPLAY & PRINT THE CURRENT STARTING CYLINDER
                    7963    ;NUMBER.
                    7964    ;
01170 FFFFF 1F17F   7965    STRTCYL:  JSR SAVRET2                    ;SAVE RETURN LOC.
01171 FFFFF 1F22B   7966              JSR STRTOVR                    ;GET STARTING CYLINDER AND...
01172 3410F F00B2   7967              ALDRAN R0,MAINILD2             ;DISPLAY IT
01173 37350 F0001   7968              RDMSKB R0,1
01174 0F322 E0E7B   7969              CALL R2,ISSCMDA1               ;MSG: "START CYL.="
01175 373F0 F00B8   7970              RDRAMB R0,STCYL                ;GET STARTING CYLINDER
01176 FFFFF E117D   7971              JMP PRTNUM                     ;PRINT START CYLINDER
                    7972
                    7973
                    7974    ;THE FOLLOWING ROUTINE WILL DISPLAY AND PRINT THE LAST ENDING CYLINDER NO.
                    7975    ;
01177 FFFFF 1F17F   7976    ENDCYL:   JSR SAVRET2                    ;SAVE RETURN LOC.
                    7977              MOVSMB   NDCYL,MAINILD3        ;Display Ending cylinder
01178 373F0 F00B9   7977 +            RDRAMB   R0,NDCYL
01179 3410F F00B3   7977 +            ALDRAN   R0,MAINILD3
0117A 37350 F0002   7978              RDMSKB R0,D#2
0117B 0F322 E0E7B   7979              CALL R2,ISSCMDA1               ;MSG: ", END CYL.="
0117C 373F0 F00B9   7980              RDRAMB R0,NDCYL                ;GET ENDING CYLINDER
                    7981    ;
                    7982    ;    The following routine prints a decimal number.
                    7983    ;The return loc. from this routine must have been previously saved
                    7984    ;with SAVRET2.
                    7985    ;The routine is called with:  JMP PRTNUM
                    7986    ;Upon entry:  R0 = Number to be printed.
                    7987    ;
0117D 0F322 E1270   7988    PRTNUM:   CALL R2,PRTNBR                 ;PRINT THE NUMBER...
0117E FFFFF E1185   7989              JMP RTRN2                      ;AND RETURN
                    7990
                    7991
0117F 3412F F01ED   7992    SAVRET2:  ALDRAM R2,RETURN2              ;SAVE RETURN-2 LSB'S
01180 3492F AF1F9   7993              ALDREGU R2,RAME & RTS RETURN2A ;SAVE RETURN-2 MSB'S
                    7994    ;
01181 3412F F01EE   7995    SAVRET3:  ALDRAM R2,RETURN3              ;SAVE RETURN-3 LSB'S
01182 3492F AF1FA   7996              ALDREGU R2,RAME & RTS RETURN3A ;SAVE RETURN-3 MSB'S
                    7997    ;
01183 3412F F01F1   7998    SAVCONT:  ALDRAM R2,ILDCONT              ;SAVE CONTINUATION LOC. LSB'S
01184 3492F AF1F0   7999              ALDREGU R2,RAME & RTS ILDCONTA ;SAVE CONTINUATION LOC. MSB'S
                    8000    ;
01185 371FE F01ED   8001    RTRN2:    MOVE RAME,IREG & NOP RETURN2
01186 371F9 F01F9   8002              MOVE RAME,IREGH & NOP RETURN2A
01187 333F0 6F7FF   8003              OR ZB,B,,R0 & JIR
                    8004    ;
01188 371FE F01EE   8005    RTRN3:    MOVE RAME,IREG & NOP RETURN3
01189 371F9 F01FA   8006              MOVE RAME,IREGH & NOP RETURN3A
0118A 333F0 6F7FF   8007              OR ZB,B,,R0 & JIR
```

Addr Line ILD SUB-ROUTINES

```
                    8009
                    8010    ;    Gets the saved flags_for_this_drive into R8.
                    8011    ;Uses:  Q & IREG.
                    8012    ;
                    8013    GETFLAG:  JSR.P FLAGLOC                    ;LOC. OF FLAGS THIS DRIVE
0118B E0401 1F342   8013 +            SETPGE FLAGLOC_SHR_N#B & JSRP FLAGLOC
0118C 373D8 AF7FF   8014              RDREGB RAM1,R8 & RTS             ;FLAGS INTO R8 & RETURN
                    8015
                    8016    ;     THE FOLLOWING ROUTINE WILL DISPLAY AND PRINT THE LAST STARTING HEAD NO.
                    8017    ;
0118D FFFFF 1F17F   8018    STRTHD:   JSR SAVRET2                      ;SAVE RETURN LOC.
0118E FFFFF 1F22D   8019              JSR STRTOVR1                     ;SET CUR. HEAD = START HEAD
0118F 373F0 F00BD   8020              RDRAMB R0,ILDNBR                 ;GET ILD NUMBER
01190 65050 F0003   8021              TSTRMSK R0,3                     ;CHECK FOR ILD 3
01191 FFFFF 33995   8022              BNZ $+4                          ;IF NOT, JUMP
                    8023    ;
01192 3411F F00B3   8024              ALDRAM R1,MAINILD3               ;DISPLAY START HEAD AS END HEAD
01193 37350 F0019   8025              RDMSKB R0,D#25                   ;MSG: ", HEAD "
01194 FFFFF E1197   8026              JMP $+3
01195 3411F F00B2   8027              ALDRAM R1,MAINILD2               ;DISPLAY START HEAD
01196 37350 F0003   8028              RDMSKB R0,D#3                    ;MSG: "START HEAD = "
01197 0F322 E0E7B   8029              CALL R2,ISSCMDA1                 ;SEND MESSAGE
01198 373F0 F00BB   8030              RDRAMB R0,STHD                   ;GET START HEAD
01199 FFFFF E117D   8031              JMP PRTNUM                       ;PRINT STARTING HEAD NO.
                    8032
                    8033
                    8034    ;    THE FOLLOWING ROUTINE WILL DISPLAY AND PRINT THE LAST ENDING HEAD NO.
                    8035    ;
0119A FFFFF 1F17F   8036    ENDHD:    JSR SAVRET2                      ;SAVE RETURN LOC.
                    8037              MOVSMB   NDHD,MAINILD3           ;Display Ending head
0119B 373F0 F00BC   8037 +            RDRAMB   R0,NDHD
0119C 3410F F00B3   8037 +            ALDRAM   R0,MAINILD3
0119D 37350 F0004   8038              RDMSKB R0,D#4
0119E 0F322 E0E7B   8039              CALL R2,ISSCMDA1                 ;MSG: ", END HEAD = "
0119F 373F0 F00BC   8040              RDRAMB R0,NDHD                   ;GET ENDING HEAD
011A0 FFFFF E117D   8041              JMP PRTNUM                       ;PRINT ENDING HEAD NO.
                    8042
                    8043
011A1 373F0 F020D   8044    HDCNT:    RDRAMB R0,MAXHDI                 ;GET MAX. ILD HEAD
011A2 FFFFF 1F279   8045              JSR CHARCNT                      ;COMPUTE MAX. NO. OF CHARACTERS
011A3 34310 AF7FF   8046              RTOR R1,R0 & RTS                 ;CHAR. COUNT TO R0 & RETURN
                    8047    ;
011A4 373F0 F01EF   8048    SETMSG1:  RDRAMB R0,MSGFLG                 ;GET THE MESSAGE FLAGS AND...
011A5 35350 F0002   8049              SETRBIT R0,MSG1                  ;SET MESSAGE SENT 1
011A6 3410F AF1EF   8050              ALDREG R0,RAME & RTS MSGFLG
                    8051    ;
011A7 373F0 F01EF   8052    CHKMSG1:  RDRAMB R0,MSGFLG                 ;GET MESSAGE FLAGS AND...
011A8 45140 AF002   8053              AND DB,FOUTL,MSK1,R0 & RTS MSG1  ;CHECK FOR MESSAGE SENT 1; RETURN
```

Addr Line ILD SUB-ROUTINES

```
                    8055
                    8056
                    8057    ;    THE FOLLOWING ROUTINE WILL REQUEST, GET AND CHECK A STARTING CYLINDER
                    8058    ;NUMBER FROM THE OPERATOR.  IF UPON ENTRY R0=0, THE DISPLAY WILL BE
                    8059    ;CLEARED BEFORE PRINTING THE MESSAGE.  IF R0=1, THE MESSAGE WILL
                    8060    ;FOLLOW ANY PRECEDING MESSAGE ON THE DISPLAY.
                    8061    ;
011A9 FFFFF 1F17F   8062    GTSTCYL:  JSR SAVRET2                   ;SAVE RETURN LOC.
011AA 3710F F00B2   8063              RESETRAM MAIN(LD2             ;CLEAR START CYLINDER DISPLAY
011AB 05340 F0007   8064              ADD DB,B,MSK1,R0 & NOP D#7
011AC 0F322 E0E7B   8065              CALL R2,ISSCMDA1              ;MSG: "ENTER STARTING CYL. NO. "
011AD FFFFF 1F1B7   8066              JSR CHCNT                     ;GET CHARS. EXPECTED
011AE 3705F F0087   8067              RDMSKQ 87
011AF 0F322 E0E7C   8068              CALL R2,ISSCMDA               ;GET START CYL. FROM THE OPERATOR
011B0 3410F F00B2   8069              ALDRAM R0,MAIN(LD2            ;DISPLAY IT
011B1 373F1 F020A   8070              RDRAMB R1,MAXCYL1             ;MAX. USABLE CYLINDER
                    8071    ;
011B2 29101 FFFFF   8072    GTSTCYL1: SUBB AB,FOUTL,R0,R1           ;IS START CYLINDER > MAX?
011B3 FFFFF 341F6   8076              CJMP AGB,BADCYL               ;IF YES, BAD ENTRY
011B4 3410F F00B8   8078              ALDRAM R0,STCYL               ;SAVE GOOD ENTRY AS START CYL...
011B5 3410F F00B5   8079              ALDRAM R0,CURCYL              ;AND AS CURRENT CYLINDER
011B6 34301 E1185   8080              RTOR R0,R1 & JMP RTRN2        ;RETURN
                    8081
                    8082
011B7 373F0 F0245   8083    CHCNT:    RDRAMB R0,MAXCYLD1            ;GET MAX. ILD DRIVE CYLINDER
011B8 FFFFF 1F279   8084              JSR CHARCNT                   ;GET MAX. NO. OF CHARACTERS
011B9 34310 AF7FF   8085              RTOR R1,R0 & RTS              ;CHAR. COUNT TO R0 & RETURN
```

Addr Line ILD SUB-ROUTINES

```
                8087
                8088
                8089    ;    THE FOLLOWING ROUTINE WILL REQUEST, GET AND CHECK AN ENDING CYLINDER
                8090    ;NUMBER FROM THE OPERATOR.  IF UPON ENTRY R0=0, CLEAR THE DISPLAY BEFORE
                8091    ;PRINTING THE MESSAGE.  IF R0=1, ADD THE MESSAGE TO THE DISPLAY.
                8092    ;
011BA FFFFF 1F17F  8093    GTNDCYL:  JSR SAVRET2                    ;SAVE RETURN LOC.
011BB 3710F F00B3  8094              RESETRAM MAINILD3              ;CLEAR END CYL. DISPLAY
011BC 05340 F0009  8095              ADD DB,B,MSK1,R0 & NOP D#9
011BD 0F322 E0E7B  8096              CALL R2,ISSCMDA1               ;MSG:  "ENTER ENDING CYL. NO. "
                8097    ;
011BE FFFFF 1F1B7  8098              JSR CHCNT                      ;GET CHARS. EXPECTED
011BF 3705F F0087  8099              RDMSKQ 87
011C0 0F322 E0E7C  8100              CALL R2,ISSCMDA                ;GET END CYL. FROM OPERATOR
011C1 3410F F00B3  8101              ALDRAM R0,MAINILD3             ;DISPLAY IT
011C2 373F1 F0245  8123              RDRAMB R1,MAXCYLD1             ;MAX. DRIVE CYLINDER
                8124    ;
011C3 29101 FFFFF  8125    GTNDCYL1: SUBB AB,FOUTL,R0,R1            ;IS END CYLINDER > MAX?
011C4 FFFFF 341F6  8129              CJMP AGB,BADCYL                ;IF IT IS, GET NEW ENTRY
011C5 3410F F00B9  8131              ALDRAM R0,NDCYL                ;SAVE ENDING CYLINDER
011C6 FFFFF E1185  8132              JMP RTRN2                      ;RETURN
```

Addr Line ILD SUB-ROUTINES

```
                    8134    ;    THE FOLLOWING ROUTINE WILL REQUEST, GET AND CHECK A STARTING (OR DEFECTIVE)
                    8135    ;HEAD NO. FROM THE OPERATOR.
                    8136    ;
011C7 FFFFF 1F17F   8137    GTSTHED:  JSR SAVRET2                       ;SAVE RETURN LOC.
011C8 373F1 F00BD   8138              RDRAMB R1,ILDNBR                  ;GET ILD NUMBER
011C9 2D041 F0003   8139              SUBB DB,0,MSK1,R1 & NOP H#3       ;CHECK FOR ILD 3 OR 4
011CA 433F0 341CD   8140              RESETR R0 & CJMP AGB,$+3          ;IF NOT, JUMP
011CB 3710F F00B3   8141              RESETRAM MAINILD3                 ;CLEAR END HEAD DISPLAY
011CC FFFFF E11CE   8142              JMP STHED1
011CD 3710F F00B2   8143              RESETRAM MAINILD2                 ;CLEAR START HEAD DISPLAY
                    8144    ;
011CE 4505F F0010   8145    STHED1:   TSTRBIT R15,10                    ;CHECK DEF. FLAG
011CF FFFFF 331D2   8146              BUZ $+3                           ;IF NOT, JUMP
011D0 05340 F000F   8147              ADD DB,B,MSK1,R0 & NOP D#15       ;MSG: "ENTER DEF. HEAD NO. "
011D1 FFFFF E11D3   8148              JMP $+2
011D2 05340 F000B   8149              ADD DB,B,MSK1,R0 & NOP D#11       ;MSG: "ENTER STARTING HEAD NO. "
011D3 0F322 E0E78   8150              CALL R2,ISSCMDA1                  ;PRINT MESSAGE
                    8151    ;
011D4 FFFFF 1F1A1   8157              JSR HDCNT                         ;$GET CHAR. COUNT
011D5 3705F F0087   8159              RDMSKQ 87
011D6 0F322 E0E7C   8160              CALL R2,ISSCMDA                   ;GET STARTING HEAD
011D7 373F1 F00BD   8161              RDRAMB R1,ILDNBR                  ;GET ILD NUMBER
011D8 2D041 F0003   8162              SUBB DB,0,MSK1,R1 & NOP H#3       ;ILD 3 OR 4?
011D9 FFFFF 341DC   8163              CJMP AGB,$+3                      ;IF NOT, JUMP
011DA 3410F F00B3   8164              ALDRAM R0,MAINILD3                ;DISPLAY START HEAD AS END HEAD
011DB FFFFF E11DD   8165              JMP $+2
011DC 3410F F00B2   8166              ALDRAM R0,MAINILD2                ;DISPLAY STARTING HEAD NUMBER
011DD 373F1 F020D   8177              RDRAMB R1,MAXHD1                  ;GET MAX. HEAD
                    8178    ;
011DE 29101 FFFFF   8179    STHED2:   SUBB AB,FOUTL,R0,R1               ;START HEAD > MAX. HEAD?
011DF FFFFF 341E4   8180              CJMP AGB,STHED4                   ;IF IT IS, GET ENTRY AGAIN
011E0 3410F F00BB   8181              ALDRAM R0,STHD                    ;SAVE STARTING HEAD
011E1 3410F F00B6   8182              ALDRAM R0,CURHD                   ;SET CUR. HEAD = START HEAD
011E2 FFFFF E1185   8183              JMP RTRN2                         ;RETURN
                    8184    ;
011E3 FFFFF 1F17F   8185    STHED3:   JSR SAVRET2
011E4 0F322 E11F3   8186    STHED4:   CALL R2,BADNTRY                   ;MSG: "BAD ENTRY. "
011E5 0F300 E11CE   8187              ADD DZC,B,ZERO,R0 & JMP STHED1    ;TRY TO GET GOOD HEAD NO.
```

Addr Line ILD SUB-ROUTINES

```
                    8189    ;    THE FOLLOWING ROUTINE WILL REQUEST, GET AND CHECK AN ENDING
                    8190    ;HEAD NO. FROM THE OPERATOR.
                    8191    ;
011E6 FFFFF 1F17F   8192    GTNDHED:  JSR SAVRET2                    ;SAVE RETURN LOC.
011E7 433F0 FFFFF   8193              RESETR R0
011E8 3710F F00B3   8194              RESETRAM MAINILD3              ;CLEAR END HEAD DISPLAY
011E9 05340 F000D   8195    NDHED1:   ADD DB,B,MSK1,R0 & NOP D#13    ;MSG: "ENTER ENDING HEAD NO. "
011EA 0F322 E0E7B   8196              CALL R2,ISSCMDA1               ;PRINT MESSAGE
                    8197    ;
011EB FFFFF 1F1A1   8203              JSR HDCNT                      ;$GET CHAR. COUNT
011EC 3705F F0087   8205              RDMSKQ 87
011ED 0F322 E0E7C   8206              CALL R2,ISSCMDA                ;GET ENDING HEAD NO.
011EE 3410F F00B3   8207              ALDRAM R0,MAINILD3             ;DISPLAY ENDING HEAD
011EF 373F1 F020D   8218              RDRAMB R1,MAXHDI               ;GET MAX. HEAD
                    8219    ;
011F0 29101 FFFFF   8220    NDHED2:   SUBB AB,FOUTL,R0,R1            ;END HEAD > MAX. HEAD?
011F1 FFFFF 341F4   8221              CJMP AGB,NDHED3                ;IF IT IS, GET NEW ENTRY
011F2 3410F F00BC   8222              ALDRAM R0,NDHD                 ;SAVE ENDING HEAD AND...
011F3 FFFFF E1185   8223              JMP RTRN2                      ;RETURN
                    8224    ;
011F4 0F322 E11F9   8225    NDHED3:   CALL R2,BADNTRY                ;MSG: "BAD ENTRY. "
011F5 0F300 E11E9   8226              ADD DZC,B,ZERO,R0 & JMP NDHED1 ;TRY TO GET GOOD END HEAD NO.
                    8227
                    8228
011F6 0F322 E11F9   8229    BADCYL:   CALL R2,BADNTRY                ;MSG: "BAD ENTRY. "
011F7 271FE F01ED   8230              DECRD RAME,IREG & NOP RETURN2  ;RETURN TO CALLING LOC. TO...
011F8 0F300 E1186   8231              ADD DZC,B,ZERO,R0 & JMP RTRN2+1 ;REQUEST NEW CYLINDER NO.
                    8232
                    8233    ;   Print "BAD ENTRY" message
                    8234    ;
011F9 37350 F0013   8235    BADNTRY:  RDMSKB R0,D#19                 ;MSG: "BAD ENTRY"
011FA FFFFF E0E7B   8236              JMP ISSCMDA1                   ;PRINT IT & RETURN
```

Addr Line ILD SUB-ROUTINES

```
                    8238
                    8239
011FB FFFFF 1F17F   8240    CONV100:  JSR SAVRET2                    ;SAVE RETURN LOC.
011FC FFFFF 1F200   8241              JSR ILDSENS                    ;SET UP SENSE BYTES
                    8242    ;
011FD 4505E F0200   8243              TSTRBIT R14,200                ;ALT. TRACK FLAG?
                    8244              CJSR.P ALUZ,CHK2NDA            ;IF NOT, CHECK FOR SECOND DRIVE
011FE E0403 13192   8244 +            SETPGE CHK2NDA_SHR_H#B & CJSRP ALUZ,CHK2NDA
011FF 433F6 E1185   8248              RESETR R6 & JMP RTRN2          ;WORD = 0 & RETURN
                    8275    ILDSENS:  EQU $
01200 0F10F F0302   8276              ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag
01201 373FB F00B5   8277              RDRAMB R11,CURCYL              ;GET THE CURRENT CYLINDER AND...
01202 373FC F00B6   8278              RDRAMB R12,CURHD               ;GET THE CURRENT HEAD
01203 0F309 E01DA   8279              ADD DZC,B,ZERO,R9 & JMP SEEKSENS ;SAVE THEN FOR SENSE; RECORD = 1
                    8280    ;
01204 FFFFF E02C0   8281    MULTPLYB: JMP MULTPLY
                    8282
                    8283
                    8284    ILDREL:   JSR.P DOUBLE                   ;DOUBLE DRIVE NO. IF NEEDED
01205 E0403 1F125   8284 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
01206 373F0 F02F4   8285              RDRAMB R0,TWODRIV              ;ANY TWO DRIVE UNITS?
                    8286              BUZ.P SELREL                   ;IF NOT, SELECT & RELEASE DRIVE
01207 E0401 33609   8286 +            SETPGE SELREL_SHR_H#B & BUZP SELREL
01208 3535D F0001   8287              SETRBIT R13,1                  ;INCREMENT TO UPPER DRIVE
                    8288              JSR.P SELREL                   ;SELECT & RELEASE UPPER DRIVE
01209 E0401 1F609   8288 +            SETPGE SELREL_SHR_H#B & JSRP SELREL
0120A 5535D F0001   8289              RSTRBIT R13,1                  ;DECREMENT TO LOWER DRIVE
0120B FFFFF E0E09   8290              JMP SELREL                     ;SELECT & RELEASE LOWER DRIVE
                    8291
                    8292
                    8293    ;    There are two entries to this routine.  They are:
                    8294    ;1) CALL R2,ILDSTRTA will print a selected message as specified in R0...
                    8295    ;2) CALL R2,ILDSTRT  will print a selected message as specified in R0,
                    8296    ;   then print ", DR.=n", where n = drive no., and then...
                    8297    ;IT WILL THEN PROMPT THE OPERATOR WITH THE QUESTION " OK?"
                    8298    ;IF THE ANSWER IS 'YES', THE ROUTINE WILL RETURN TO CALL +1.
                    8299    ;IF THE ANSWER IS 'NO', THE ILD WILL BE TERMINATED (ILD28).
                    8300    ;
0120C FFFFF 1F181   8301    ILDSTRT:  JSR SAVRET3                    ;SAVE THE RETURN LOC.
0120D 0F322 E0E7B   8302              CALL R2,ISSCMDA1               ;PRINT THE MESSAGE
0120E 433F0 FFFFF   8303              RESETR R0
0120F 0F322 E0E7B   8304              CALL R2,ISSCMDA1               ;MSG: ", DR.="
01210 0F322 E126E   8305              CALL R2,PRNTDRV                ;PRINT THE DRIVE NO.
01211 0F322 E125B   8306    ILDSTRT1: CALL R2,YESNO                  ;IS THIS O.K?
01212 FFFFF 33988   8307              BNZ RTRN3                      ;IF IT IS, RETURN
```

```
 Addr            Line ILD SUB-ROUTINES

01213 FFFFF E1133  8308            JMP ILD28                    ;ELSE, TERMINATE THE ILD
                   8309    ;
01214 FFFFF 1F181  8310    ILDSTRTA: JSR SAVRET3                 ;SAVE THE RETURN LOC.
01215 0F322 E0E78  8311            CALL R2,ISSCMDA1             ;PRINT THE MESSAGE
01216 FFFFF E1211  8312            JMP ILDSTRT1
```

Addr Line ILD SUB-ROUTINES

```
                    8314
                    8315
                    8316    ;    THE FOLLOWING ROUTINE WILL CHECK TO SEE IF THE OPERATOR WANTS TO HALT
                    8317    ;AN ILD.  IF NOT, THE ROUTINE WILL RETURN.
                    8318    ;    ONCE HALTED, THE ILD MAY BE TERMINATED OR CONTINUED DEPENDING ON THE ANSWER
                    8319    ;TO THE PRINTED QUESTION "RESTART?"
                    8320    ;    THE ILD IS HALTED BY:  ILD-0 OR LOAD-PB-80.
                    8321    ;    UPON EXIT:  R0=0 IF ILD WAS NOT HALTED.  R0=1 IF ILD WAS HALTED AND
                    8322    ;RESTARTED.
                    8323    ;
                    8324    ;
01217 FFFFF 1F17F   8325    ILDHALT:  JSR SAVRET2                       ;SAVE RETURN LOC.
01218 C3000 FFFFF   8326              RD SCP,PRGCPB,R0                  ;READ PUSH-BUTTON REG.
01219 C3000 FFFFF   8327              RD SCP,PRGCPB,R0
0121A 65050 F0080   8328              TSTRMSK R0,80                     ;CHECK FOR PB = 80
0121B 47360 33985   8329              AND DZ,B,SMCST,R0 & BNZ RTRN2     ;IF NOT, CONTINUE
                    8330    ;
0121C 5535F F8400   8331              RSTRBIT R15,8400                  ;RESET LAST REC./SM RETRY FLAGS
0121D E30C0 1F205   8332              CTL SCP,MPPGCLR & JSR ILDREL      ;RELEASE & DE-SELECT DRIVE
0121E 3710F F01EF   8333              RESETRAM MSGFLG                   ;RESET MESSAGE SENT & LOOP FLAGS
0121F 37350 F001B   8334              RDMSKB R0,D#27
01220 0F322 E0E7B   8335              CALL R2,ISSCMDA1                  ;MSG: "ILD HALTED @ CYL. "
01221 373FE F01F2   8336              RDRAMB R14,R14SAVL                ;RESTORE R14
01222 0F322 E128D   8337              CALL R2,CYLHED1                   ;DISPLAY CUR. CYL. & HEAD
01223 37350 F001D   8338              RDMSKB R0,D#29
01224 0F322 E0E7B   8339              CALL R2,ISSCMDA1                  ;MSG: "RESTART?"
                    8340    ;
01225 3705F F0085   8341              RDMSKQ 85
01226 0F322 E0E7C   8342              CALL R2,ISSCMDA                   ;WAIT FOR 'YES' OR 'NO'
01227 FFFFF 33133   8343              BOZ ILD28                         ;IF 'NO', TERMINATE THE ILD
01228 3710F F0018   8344              RESETRAM SEN415                   ;RESET RETRY COUNT
01229 FFFFF 1F233   8345              JSR ILDDSP                        ;DISPLAY ILD & DRIVE NUMBERS
0122A FFFFF E1185   8346              JMP RTRN2                         ;RESTART THE TEST
                    8347
                    8348
                    8349    ;    THIS ROUTINE WILL SET THE CURRENT CYLINDER & HEAD NUMBERS TO THE
                    8350    ;VALUE OF THE STARTING CYLINDER & HEAD NUMBERS.
                    8351    ;
                    8352    STRTOVR:  MOVSMB   STCYL,CURCYL             ;Set Cur. cyl.= Start cyl.
0122B 373F0 F00B8   8352 +            RDRAMB   R0,STCYL
0122C 3410F F00B5   8352 +            ALDRAM   R0,CURCYL
0122D 373F1 F00BB   8353    STRTOVR1: RDRAMB R1,STHD                    ;GET STARTING HEAD
0122E 3411F AF0B6   8354              ALDREG R1,RAME & RTS CURHD        ;CUR. HEAD = START HEAD & RETURN
```

Addr Line ILD SUB-ROUTINES

```
                    8356
                    8357    ;    THIS SUBROUTINE WILL GET THE CURRENT HEAD INTO R12 AND SAVE IT AS THE
                    8358    ;NEXT HEAD.  THE CURRENT RECORD (R9) AND THE NEXT RECORD WILL BE SET TO 1.
                    8359    ;
0122F 373FC F00B6   8360    RECORD1:  RDRAMB R12,CURHD                  ;GET THE CURRENT HEAD NO.
01230 341CF F0027   8361              ALDRAM R12,NEXTHD                 ;SAVE IT AS THE NEXT HEAD
01231 0F309 FFFFF   8362              ADD DZC,B,ZERO,R9                 ;RECORD NO.= 1
01232 3419F AF028   8363              ALDREG R9,RAME & RTS NEXTREC      ;SAVE NEXT RECORD & RETURN
                    8364    ;
01233 373F0 F00BD   8365    ILDDSP:   RDRAMB R0,ILDNBR                  ;GET THE ILD NUMBER
01234 01700 4F002   8366              ADD AB,LSBZ,R0,R0 & PSHLD 2
                    8367              LSHFT2 R0,R0                      ;MOVE IT INTO POSITION
01235 01700 887FF   8367 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01236 370FF F00B4   8368              RDRAMQ ILDDRV                     ;GET THE DRIVE NUMBER
01237 4605F F003F   8369              TSTQBIT 3F                        ;DELETE VALID BIT
01238 3010F AF0B1   8370              OR AQ,FOUTL,R0,RAME & RTS MAINILD1 ;DISPLAY ILD & DR. NOS.; RETURN
                    8371
                    8372
                    8373    ;    THE FOLLOWING ROUTINE WILL CHECK FOR AN ILD RUNNING AND THE DRIVE
                    8374    ;SELECTED AND READY.  IF ALL OF THESE CONDITIONS ARE NOT MET, THE PROGRAM
                    8375    ;WILL ATTEMPT TO RE-SELECT THE ILD DRIVE.
                    8376    ;UPON EXIT:  R3  = ILD FLAGS
                    8377    ;            R13 = PHYSICAL DRIVE NO.
                    8378    ;
01239 FFFFF 1F17F   8379    ILDRUN:   JSR SAVRET2                       ;SAVE RETURN LOC.
0123A 373FD F00B4   8380              RDRAMB R13,ILDDRV                 ;GET ILD DRIVE NUMBER
0123B 5535D F0800   8381              RSTRBIT R13,VALID                 ;DELETE THE VALID BIT
0123C 373F3 F00BA   8382              RDRAMB R3,ILDFLG                  ;GET ILD FLAGS
0123D 45053 F0004   8388              TSTRBIT R3,4                      ;CHECK FOR ILD RUNNING
0123E FFFFF 33248   8389              BUZ ILDRUN2                       ;IF NOT, DRIVE MUST BE RE-SELECTED
                    8390    ;
0123F C1041 FFFFF   8392              RD       HSSM,RDDRST1,R1          ;$
01240 C1041 FFFFF   8398              RD HSSM,RDDRST1,R1                ;READ DRIVE STATUS
01241 45051 F002D   8399              TSTRBIT R1,SELECT+READY+FAULT+SEEKERR ;SELECTED, READY & ERRORS
01242 6605F F0021   8400              TSTQMSK SELECT+READY              ;SELECTED & READY ONLY?
01243 FFFFF 33A48   8401              BNZ ILDRUN2                       ;IF NOT, JUMP
                    8402    ;
                    8403              JSR.P DOUBLE                      ;DOUBLE DRIVE NO. IF NEEDED
01244 E0403 1F125   8403 +            SETPGE DOUBLE_SHR_M#B & JSRP DOUBLE
01245 4505E F0800   8404              TSTRBIT R14,800                   ;SECOND DRIVE FLAG?
01246 FFFFF 33185   8405              BUZ RTRN2                         ;IF NOT, RETURN
01247 0B3FD E1185   8406              INCRB R13 & JMP RTRN2             ;INCR. DRIVE NO. & RETURN
                    8407    ;
01248 35353 F0004   8408    ILDRUN2:  SETRBIT R3,4                      ;SET ILD RUNNING FLAG
01249 3413F F00BA   8409              ALDRAM R3,ILDFLG
0124A 373F1 F00BD   8410              RDRAMB R1,ILDNBR
```

```
 Addr            Line ILD SUB-ROUTINES

                 8417   ;
                 8418            JSR.P SEL00                     ;SELECT ILD DRIVE
0124B E0401 1F169 8418 +         SETPGE SEL00_SHR_H#B & JSRP SEL00
0124C 373FB F00B5 8419           RDRANB R11,CURCYL               ;GET CURRENT CYLINDER
0124D 373FC F00B6 8420           RDRANB R12,CURHD                ;GET CURRENT HEAD
0124E 373F3 F00BA 8421           RDRANB R3,ILDFLG                ;ILD FLAGS INTO R3
0124F 0F309 E1185 8422           ADD DZC,B,ZERO,R9 & JMP RTRN2   ;RECORD NO.= 1 & RETURN
                 8423
                 8424   ;    THIS ROUTINE WILL WRITE ONE TRACK TO THE ILD DRIVE.
                 8425   ;THE ROUTINE IS ENTERED WITH:  CALL R2,ILDWRT1
                 8426   ;THE REQUIREMENTS FOR THIS ROUTINE ARE:
                 8427   ; 1.) THE ILD DRIVE MUST BE SELECTED.
                 8428   ; 2.) THE MDB MUST CONTAIN THE DATA TO BE WRITTEN.
                 8429   ; 3.) CURCYL & CURHD MUST CONTAIN THE CYLINDER AND HEAD TO BE WRITTEN.
                 8430   ;
01250 377F7 F02D9 8431   ILDWRT1:  OR DZ,LSBZ,RAME,R7 & NOP WRDSTRK
01251 337F7 FFFFF 8432           OR ZB,LSBZ,,R7                  ;WORDS TO BE WRITTEN TO R7
01252 FFFFF 1F183 8433   ILDWRT2:  JSR SAVCONT                   ;SAVE THE RETURN LOC.
01253 433FF 1F200 8434           RESETR R15 & JSR ILDSENS        ;SET UP SENSE BYTES FOR SEEK
                 8435            JSR.P SEEK02                    ;POSITION THE DRIVE
01254 E0400 1F12A 8435 +         SETPGE SEEK02_SHR_H#B & JSRP SEEK02
01255 5535E FFBDF 8436           RSTRBIT R14,H#FBDF              ;RESET SOME FLAGS
01256 3535E F0400 8437           SETRBIT R14,400                 ;SET U-COMPLETE FLAG
01257 35355 F0800 8438           SETRBIT R5,800                  ;SET WRITE-OP. FLAG
01258 433F6 FFFFF 8439           RESETR R6                       ;FIRST WORD IS 0
                 8440            JSR.P RDWRTCLR                  ;CLEAR THE DISK BOARDS AND...
01259 E0400 1F4B7 8440 +         SETPGE RDWRTCLR_SHR_H#B & JSRP RDWRTCLR
0125A 433F4 E05AF 8441           RESETR R4 & JMP WRT035          ;WRITE ONE TRACK
                 8442
                 8443
                 8444   ;    THE FOLLOWING ROUTINE WILL SOLICIT A "YES" OR "NO" RESPONSE FROM THE
                 8445   ;OPERATOR.  UPON EXIT: R0 = 0 FOR "NO", R0 = 1 FOR "YES".
                 8446   ;The routine YESNO asks the question, "OK?".
                 8447   ;The routine YESNOA just waits for a YES or NO response.
                 8448   ;
0125B FFFFF 1F17F 8449   YESNO:    JSR SAVRET2                   ;SAVE RETURN LOC.
0125C 37350 F002F 8450           RDMSKB R0,D#47
0125D 0F322 E0E7B 8451           CALL R2,ISSCMDA1                ;MSG: " OK?"
0125E 3705F F0085 8452   YESNO1:   RDMSKQ 85                     ;COMMAND FOR 'YES' OR 'NO'
0125F 0F322 E0E7C 8453           CALL R2,ISSCMDA                 ;WAIT FOR OPERATOR RESPONSE
                 8454   ;
01260 FFFFF E1185 8455           JMP RTRN2                       ;RETURN
                 8456   ;
01261 FFFFF 1F17F 8457   YESNOA:  JSR SAVRET2                   ;SAVE RETURN LOC.
01262 FFFFF E125E 8458           JMP YESNO1
```

Addr Line ILD SUB-ROUTINES

```
                    8460    ;    THE FOLLOWING ROUTINE WILL DISPLAY IN THE LIGHTS AND SAVE THE CURRENT
                    8461    ;HEAD AND CYLINDER NUMBERS THAT ARE IN R12 & R11 RESPECTIVELY.  THE CURRENT
                    8462    ;CYLINDER NO. WILL BE PRINTED.
                    8463    ;
01263 FFFFF 1F181   8464    DSPLY:    JSR SAVRET3                    ;SAVE RETURN LOC.
01264 FFFFF 1F280   8465              JSR ILDCYHD                    ;DISPLAY & SAVE CUR. HEAD & CYL.
01265 37350 F0017   8466              RDMSKB R0,D#23                 ;CURSOR POSITION
                    8467    ;
01266 3705F F010A   8468              RDMSKQ 10A                     ;POSITION CURSOR & WAIT COMMAND
01267 0F322 E0E7C   8469              CALL R2,ISSCMDA                ;POSITION THE DISPLAY CURSOR
01268 373F0 F00B5   8470              RDRAMB R0,CURCYL               ;GET THE CURRENT CYLINDER...
01269 0F322 E1270   8471              CALL R2,PRTNBR                 ;AND PRINT IT
0126A FFFFF E1180   8472              JMP RTRN3                      ;RETURN
                    8473
                    8474
0126B 3710F F006F   8475    RST1BIT:  RESETRAM BITJLD                ;RESET THE...
0126C 3710F F0070   8476              RESETRAM BITILD+1              ;BIT FOR THE...
0126D 3710F AF071   8477              MOVE ZERO,RAME & RTS BITJLD+H#2 ;ILD DRIVE
                    8478
                    8479    ;    THIS ROUTINE WILL PRINT THE CURRENT JLD DRIVE NUMBER.
                    8480    ;IT IS ENTERED WITH:  CALL R2,PRNTDRV
                    8481    ;
0126E 373F0 F00B4   8482    PRNTDRV:  RDRAMB R0,ILDDRV               ;GET THE ILD DRIVE NUMBER
0126F 55350 F0880   8483              RSTRBIT R0,VALID+H#80          ;DELETE VALID BIT & S.S.S. BIT
                    8484
                    8485    ;    THIS ROUTINE PRINTS A DECIMAL NUMBER.
                    8486    ;UPON ENTRY:  R0 = NUMBER TO BE PRINTED
                    8487    ;             R2 = RETURN LOCATION
                    8488    ;
01270 FFFFF 1F279   8489    PRTNBR:   JSR CHARCNT                    ;COMPUTE CHAR. COUNT
01271 3705F F0009   8490              RDMSKQ 9
01272 FFFFF E0E7C   8491              JMP ISSCMDA                    ;PRINT A DECIMAL NUMBER
                    8492    ;
01273 C3060 FFFFF   8493    FRNTDAT:  RD SCP,DATSW3,R0               ;GET DATA BITS 24-35 AND...
01274 3410F F026A   8494              ALDRAM R0,FORDAT3              ;SAVE THEM
01275 C3050 FFFFF   8495              RD SCP,DATSW2,R0               ;GET DATA BITS 12-23 AND...
01276 3410F F026B   8496              ALDRAM R0,FORDAT2              ;SAVE THEM
01277 C3040 FFFFF   8497              RD SCP,DATSW1,R0               ;GET DATA BITS 0-11 AND...
01278 3410F AF26C   8498              ALDREG R0,RAME & RTS FORDAT1   ;SAVE THEM; RETURN
```

Addr Line ILD SUB-ROUTINES

```
                    8500    ;
01279 2D150 F000A   8501    CHARCNT:  SUBB DB,FOUTL,MSK2,R0 & NOP D#10   ;IS THE NUMBER > 10?
0127A 0F301 A47FF   8502              ADD DZC,B,ZERO,R1 & CRTS AGB       ;IF NOT, COUNT = 1
0127B 2D150 F0064   8503              SUBB DB,FOUTL,MSK2,R0 & NOP D#100  ;IS THE NUMBER > 100?
0127C 0B3F1 A47FF   8504              INCRB R1 & CRTS AGB                ;IF NOT, COUNT = 2
0127D 2D150 F03E8   8505              SUBB DB,FOUTL,MSK2,R0 & NOP D#1000 ;IS THE NUMBER > 1000?
0127E 0B3F1 A47FF   8506              INCRB R1 & CRTS AGB                ;IF NOT, COUNT = 3
0127F 0B3F1 AF7FF   8507              INCRB R1 & RTS                     ;ELSE COUNT = 4
                    8508
                    8509
                    8510    ;    THIS ROUTINE WILL DISPLAY THE CURRENT CYLINDER AND HEAD IN THE
                    8511    ;LIGHTS (DISPLAY 43B).  IT THEN SETS THE CURRENT CYLINDER AND HEAD
                    8512    ;TO THE NEXT CYLINDER (R11) AND HEAD (R12).
                    8513    ;
01280 370FF F00B6   8514    ILDCYHD:  RDRAMQ CURHD                       ;GET CURRENT HEAD AND...
01281 321FF F00B3   8515              QLDRAM MAINILD3                    ;DISPLAY IT
01282 373F1 F00B5   8516              RDRAMB R1,CURCYL                   ;GET CURRENT CYLINDER AND...
01283 3411F F00B2   8517              ALDRAM R1,MAINILD2                 ;DISPLAY IT
                    8518    ;
01284 C3000 FFFFF   8519              RD SCP,PRGCPB,R0                   ;READ PUSH-BUTTON REG.
01285 65050 F0080   8520              TSTRMSK R0,80                      ;CHECK FOR ILD HALT
01286 FFFFF A37FF   8521              CRTS ALUZ                          ;IF IT IS, RETURN
                    8522    ;
01287 341CF F00B6   8523    ILDCYHD1: ALDRAM R12,CURHD                   ;SAVE CUR. HEAD AS NEXT HEAD
01288 341BF AF0B5   8529              ALDREG R11,RAME & RTS CURCYL       ;SAVE CUR. CYLINDER & RETURN
```

Addr Line ILD SUB-ROUTINES

```
                    8531
                    8532
                    8533    ;    THE FOLLOWING ROUTINE WILL PRINT THE CURRENT CYLINDER & HEAD NUMBERS.
                    8534    ;
01289 FFFFF 1F2A3   8535    CYLHED:   JSR SAVRET4                     ;SAVE RETURN LOC.
0128A 37350 F001C   8536              RDMSKB R0,D#28
0128B 0F322 E0E7B   8537              CALL R2,ISSCMDA1                ;MSG: " CYL. "
0128C FFFFF E128E   8538              JMP $+2
                    8539    ;
0128D FFFFF 1F2A3   8540    CYLHED1:  JSR SAVRET4                     ;SAVE RETURN LOC.
0128E 373F0 F00B5   8541              RDRAMB R0,CURCYL                ;GET CURRENT CYLINDER
0128F 0F322 E1270   8542              CALL R2,PRTNBR                  ;PRINT CYLINDER NO.
                    8543    ;
01290 373F1 F00BD   8544              RDRAMB R1,ILDNBR                ;GET CUR. ILD NO.
01291 2D141 F0005   8545              SUBB DB,FOUTL,MSK1,R1 & NOP H#5 ;IF THE ILD NO. IS...
01292 FFFFF 3429C   8546              CJMP AGB,CYLHED3                ;LESS THAN 5, PRINT HEAD NO.
                    8547    ;
01293 4505F F8000   8548              TSTRBIT R15,8000                ;SEEK ERROR FLAG?
01294 603FF 3329A   8549              XOR AQ,B,R15,R15 & BUZ CYLHED2  ;IF YES, RESET IT; IF NO, JUMP
01295 37350 F0052   8550              RDMSKB R0,D#82
01296 0F322 E0E7B   8551              CALL R2,ISSCMDA1                ;MSG: ", BAD="
01297 373F0 F00B3   8552              RDRAMB R0,MAINILD3              ;GET BAD CYL. AND...
01298 0F322 E1270   8553              CALL R2,PRTNBR                  ;PRINT IT
01299 FFFFF E12A0   8554              JMP RTRN4                       ;RETURN
                    8555    ;
0129A 2D141 F000B   8556    CYLHED2:  SUBB DB,FOUTL,MSK1,R1 & NOP H#B ;IF THE ILD NO. IS...
0129B FFFFF 342A0   8557              CJMP AGB,RTRN4                  ;LESS THAN 'B', RETURN
                    8558    ;
0129C 37350 F0019   8559    CYLHED3:  RDMSKB R0,D#25
0129D 0F322 E0E7B   8560              CALL R2,ISSCMDA1                ;MSG: ", HEAD "
0129E 373F0 F00B6   8561              RDRAMB R0,CURHD                 ;GET CUR. HEAD
0129F 0F322 E1270   8562              CALL R2,PRTNBR                  ;PRINT HEAD NO.
012A0 371FE F023C   8563    RTRN4:    MOVE RAME,IREG & NOP RETURN4    ;GET RETURN LOC. LSB'S
012A1 371F9 F023B   8564              MOVE RAME,IREGH & NOP RETURN4A  ;AND MSB
012A2 FFFFF 6F7FF   8565              JIR                             ;RETURN
                    8566    ;
012A3 3412F F023C   8567    SAVRET4:  ALDRAM R2,RETURN4
012A4 3492F AF23B   8568              ALDREGU R2,RAME & RTS RETURN4A
```

Addr Line ILD SUB-ROUTINES

```
                    8570
                    8571
                    8572    ;    THE FOLLOWING ROUTINE HANDLES THE ILD DRIVE BEING WRITE PROTECTED WHEN
                    8573    ;TRYING TO DO A WRITING TYPE TEST.  THE ROUTINE WILL NOT RETURN UNTIL "WRITE
                    8574    ;PROTECT" IS TURNED OFF OR THE ILD IS HALTED (LOAD-PB-8-0 OR ILD-0).
                    8575    ;
012A5 FFFFF 1F181   8576    WRTPROT:  JSR SAVRET3                       ;SAVE RETURN LOC.
012A6 37350 F0027   8577              RDMSKB R0,D#39                    ;MSG: "DRIVE IN WRITE PROTECT"
012A7 0F322 E0E7B   8578              CALL R2,ISSCMDA1
                    8579    ;
012A8 370FF F01EF   8580              RDRAMQ MSGFLG                     ;GET MSG. SENT FLAGS
012A9 5605F F0004   8581              RSTQBIT MSG2                      ;CLEAR MSG. SENT 2 AND...
012AA 3605F F0002   8582              SETQBIT MSG1                      ;SET MSG. SENT 1 FLAG
012AB 321FF F01EF   8583              QLDRAM MSGFLG
                    8584    ;
012AC 37350 F03E8   8585    WRTPROT1: RDMSKB R0,D#1000                  ;SET PASS COUNTER FOR 1000 PASSES
012AD 37352 F12AF   8586              RDMSKB R2,$+2                     ;RETURN LOC. FROM THE WAIT LOOP
012AE FFFFF E10B6   8587              JMP ILD10A+1                      ;GO THROUGH THE WAIT LOOP
                    8588    ;
                    8589    ;  RETURN FROM WAIT LOOP
                    8590              JSR.P WAIT102                     ;RESTORE REGISTERS
012AF E0402 1F002   8590 +            SETPGE WAIT102_SHR_H#B & JSRP WAIT102
012B0 373F0 F01FC   8591              RDRAMB R0,PASCNT                  ;GET THE PASS COUNTER
012B1 FFFFF 332B8   8592              BOZ WRTPROT2                      ;IF 0, CHECK AGAIN FOR WRITE PROT.
012B2 1410F F01FC   8593              SBA AZ,FOUTL,R0,RAME & NOP PASCNT ;DECR. PASS COUNTER
012B3 0F322 E1217   8594              CALL R2,ILDHALT                   ;CHECK FOR OPERATOR STOPPING ILD
                    8595              BOZ.P WAIT100                     ;IF NOT, ONCE MORE THRU WAIT LOOP
012B4 E0402 3300F   8595 +            SETPGE WAIT100_SHR_H#B & BOZP WAIT100
                    8596    ;
                    8597              RSTMBIT MSG2+MSG1,MSGFLG          ;Reset Msg. Sent flags
012B5 373F0 F01EF   8597 +            RDRAMB   R0,MSGFLG
012B6 55350 F0006   8597 +            RSTRBIT  R0,MSG2+MSG1
012B7 3410F F01EF   8597 +            ALDRAM   R0,MSGFLG
                    8598    ;
                    8599    WRTPROT2: JSR.P RELDRIV                     ;RELEASE & DE-SELECT DRIVE
012B8 E0401 1F60F   8599 +            SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
012B9 5535F F0100   8600              RSTRBIT R15,100                   ;RESET READ ONLY FLAG
012BA 0F322 E1239   8601              CALL R2,ILDRUN                    ;RE-SELECT THE DRIVE
012BB 4505F F0100   8602              TSTRBIT R15,100                   ;CHECK FOR 'WRITE PROTECT'
012BC FFFFF 332C1   8603              BOZ WRTPROT3                      ;IF NOT, JUMP
                    8604    ;
                    8605              TSTMBIT MSG1,MSGFLG               ;Check Msg. Sent 1
012BD 373F0 F01EF   8605 +            RDRAMB   R0,MSGFLG
012BE 45150 F0002   8605 +            TSTBIT   R0,MSG1
012BF FFFFF 332A6   8606              BOZ WRTPROT+1                     ;IF NOT, REPRINT WRITE PROT. MSG.
012C0 FFFFF E12AC   8607              JMP WRTPROT1                      ;CONTINUE WAITING
                    8608    ;
012C1 3705F F000F   8609    WRTPROT3: RDMSKQ H#F
012C2 0F322 E0E7C   8610              CALL R2,ISSCMDA                   ;CLEAR MESSAGE DISPLAY
```

Addr Line ILD SUB-ROUTINES

```
                8611            RSTMBIT MSG2+MSG1,MSGFLG        ;Reset Msg. Sent flags
012C3 373F0 F01EF 8611 +        RDRAMB  R0,MSGFLG
012C4 55350 F0006 8611 +        RSTRBIT R0,MSG2+MSG1
012C5 3410F F01EF 8611 +        ALDRAM  R0,MSGFLG
012C6 0F322 E1239 8612          CALL R2,ILDRUN                  ;RE-SELECT DRIVE IF NECESSARY
012C7 37350 F10D2 8613          RDMSKB R0,ILD20                 ;SAVE SM-ERROR RETURN LOC...
                8614            JSR.P SAVSMRTN                  ;BEFORE CONTINUING ILD
012C8 E0400 1F4C6 8614 +        SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
012C9 FFFFF E1188 8615          JMP RTRN3                       ;RETURN TO ILD
                8616
                8617    ;    THIS ROUTINE WILL CHECK IF THERE IS A FLAG BYTE FOR THIS DRIVE.  IF THERE
                8618    ;IS, CONSTANTS AND LIMITS WILL BE SET UP (USING SEL10).  IF NOT, WE WILL
                8619    ;SELECT THE DRIVE IN ORDER TO READ THE FLAG BYTE.  IF THE FLAG BYTE CANNOT
                8620    ;BE READ (EXCEPT FOR ILD-1), THE MESSAGE "DR. __ NEEDS FORMATTING. ILD
                8621    ;TERMINATED" WILL BE PRINTED AND THE ILD WILL BE TERMINATED.
                8622    ;    IF A HEADER CANNOT BE READ AND YOU WANT TO RUN AN ILD (EXCEPT FORMAT),
                8623    ;LOAD '8N' INTO THE TOGGLE REGISTER BEFORE STARTING THE ILD.  N = A NUMBER
                8624    ;BETWEEN 1 & 7 WHICH REPRESENTS THE DRIVE TYPE BEING TESTED.
                8625    ;
012CA FFFFF 1F181 8626  CKFLAG:  JSR SAVRET3                    ;SAVE RETURN LOC.
012CB FFFFF 1F188 8632          JSR GETFLAG$                    ;GET FLAG BYTE
012CC FFFFF 332D1 8633          BOZ CKFLAG1                     ;IF NONE, JUMP
012CD 370FF F03E5 8634          RDRAMQ LASTFLG                  ;GET LAST DRIVE FLAG
012CE FFFFF 332D1 8635          BOZ CKFLAG1                     ;IF NONE, JUMP
                8636    CKFLAGA: JSR.P SEL10A                   ;ELSE, SET UP CONSTANTS & LIMITS
012CF E0401 1F1E5 8636 +        SETPGE SEL10A_SHR_H#B & JSRP SEL10A
012D0 34380 E1188 8637          RTOR R8,R0 & JMP RTRN3          ;RETURN WITH FLAGS IN R0
                8638    ;
012D1 37352 F12E9 8639  CKFLAG1: RDMSKB R2,CKFLAG5              ;LOOP-ON-ERROR LOCATION
012D2 FFFFF 1F183 8640          JSR SAVCONT                     ;SAVE IT IN CASE OF ERROR
012D3 0F322 E1239 8641  CKFLAG2: CALL R2,ILDRUN                 ;SELECT THE ILD DRIVE
012D4 5535E F0200 8642          RSTRBIT R14,200                 ;RESET ALT. TRACK FLAG
012D5 FFFFF 1F18B 8643          JSR GETFLAG$                    ;GET SAVED FLAGS
012D6 34380 332D9 8644          RTOR R8,R0 & BOZ CKFLAG3        ;IF NO FLAGS, JUMP
                8645            JSR.P RELDRIV                   ;RELEASE & DE-SELECT THE DRIVE
012D7 E0401 1F60F 8645 +        SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
012D8 FFFFF E1188 8646          JMP RTRN3                       ;RETURN WITH FLAGS IN R0
                8647    ;
                8648    ;    UNABLE TO DETERMINE DRIVE TYPE
                8649    ;
012D9 370FF F00BD 8650  CKFLAG3: RDRAMQ ILDNBR                  ;GET THE ILD NUMBER
012DA 6605F F0001 8651          TSTQMSK 1                       ;IS IT ILD-1 (FORMAT)
012DB C3011 33188 8652          RD SCP,PRGTGL,R1 & BOZ RTRN3    ;IF YES, RETURN; READ TOGGLE REG.
                8653    ;
012DC 45051 F0080 8654          TSTRBIT R1,80                   ;FORCE THE HEADER?
012DD 60311 332E2 8655          XOR AQ,B,R1,R1 & BOZ CKFLAG4    ;IF NOT, JUMP
012DE 45741 F0007 8656          AND DB,LSBZ,MSK1,R1 & NOP H#7   ;ANY DRIVE TYPE?
012DF FFFFF 332E2 8657          BOZ CKFLAG4                     ;IF NOT, JUMP
```

```
 Addr            Line ILD SUB-ROUTINES

012E0 35351 F0010  8659            SETRBIT R1,10                    ;$PREP. LENGTH = 116 WORDS
012E1 3411D E12CF  8660            ALDREG R1,RAMI & JMP CKFLAGA     ;$SAVE IT & JUMP
                   8664    ;
012E2 37350 F0058  8665    CKFLAG4: RDMSKB R0,D#88                   ;SEND MESSAGE:...
012E3 0F322 E0E7B  8666            CALL R2,ISSCMDA1                 ;"DR. "
012E4 0F322 E126E  8667            CALL R2,PRNTDRV                  ;PRINT DRIVE NUMBER
012E5 37350 F0059  8668            RDMSKB R0,D#89
012E6 0F322 E0E7B  8669            CALL R2,ISSCMDA1                 ;MSG: " NEEDS FORMATTING. "
012E7 37350 F0024  8670            RDMSKB R0,D#36
012E8 FFFFF E1134  8671            JMP ILD28+1                      ;TERMINATE THE ILD
                   8672    ;
                   8673    CKFLAG5: JSR.P RELDRIV                   ;RELEASE & DE-SELECT THE DRIVE
012E9 E0401 1F60F  8673 +          SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
012EA FFFFF E12D3  8674            JMP CKFLAG2                      ;TRY AGAIN
                   8675
                   8676    ;    THIS ROUTINE WILL CHECK FOR MAX. CYLINDER AND HEAD.  IF NOT MAX., THE
                   8677    ;ROUTINE WILL RETURN WITH ALU=0.  IF MAX. CYL. & HEAD, ALU=1 ON RETURN.
                   8678    ;
                   8679    ;IF NOT MAX., UPON EXIT:
                   8680    ;R9  = NEXT RECORD
                   8681    ;R11 = NEXT CYLINDER
                   8682    ;R12 = NEXT HEAD
                   8683    ;
012EB 370FF F00B5  8684    LIMIT:   RDRAMQ CURCYL                   ;GET CURRENT CYLINDER
012EC 373F1 F00B6  8685            RDRAMB R1,CURHD                  ;GET CURRENT HEAD
012ED 2E1F0 F00B9  8686            SUBB DQ,FOUTL,RAME,ZERO & NOP NDCYL ;IF CUR. CYL. IS LESS...
012EE FFFFF 342F5  8687            CJMP AGB,LIMITB                  ;THAN ENDING CYL., CONTINUE
012EF 2E1F0 F00B9  8688            SUBB DQ,FOUTL,RAME,ZERO & NOP NDCYL ;IF CUR. CYL. IS GREATER THAN...
012F0 FFFFF 352F3  8689            CJMP ALB,LIMITA                  ;END CYL., MAX. LIMIT HAS PASSED
                   8690    ;
012F1 2D1F1 F00BC  8691            SUBB DB,FOUTL,RAME,R1 & NOP NDHD ;IF CUR. HEAD NO. IS LESS THAN...
012F2 FFFFF 342F5  8692            CJMP AGB,LIMITB                  ;ENDING HEAD. NO., CONTINUE
                   8693    ;
012F3 5535F F1800  8701    LIMITA:  RSTRBIT R15,1800                 ;RESET ADV CYL./HD. FLAGS
012F4 0F100 AF7FF  8703            ADD DZC,FOUTL,ZERO,ZERO & RTS    ;RETURN WITH ALU = 1
                   8704    ;
012F5 373FB F0026  8705    LIMITB:  RDRAMB R11,NEXTCYL               ;GET NEXT CYLINDER
012F6 373FC F0027  8706            RDRAMB R12,NEXTHD                ;GET NEXT HEAD
012F7 373F9 F0028  8707            RDRAMB R9,NEXTREC                ;GET NEXT RECORD
012F8 431F0 AF7FF  8708            AND ZB,FOUTL,,ZERO & RTS         ;RETURN WITH ALU = 0
```

Addr Line ILD SUB-ROUTINES

```
                    8710    ;    SPECIAL DRIVE SELECTION ROUTINE
                    8711    ;    IF THE DRIVE IS SUCCESSFULLY SELECTED THE PROGRAM WILL RTS BACK.
                    8712    ;IF THE DRIVE IS NOT READY, THE PROGRAM WILL JUMP TO AN ERROR EXIT LOCATION
                    8713    ;THAT WAS PREVIOUSLY STORED IN 'RETURN5'.  THIS ROUTINE WILL NOT DETERMINE
                    8714    ;THE DRIVE TYPE.
                    8715    ;
012F9 3492F F02FE   8716    SAVRET5:  ALDRAMU R2,RETURN5A               ;SAVE RETURN MSB AND...
012FA 3412F AF2FF   8717              ALDREG R2,RAME & RTS RETURN5      ;THE LSB'S
                    8718    ;
012FB 370FF F00BA   8719    SELDRIV:  RDRAMQ ILDFLG                     ;GET ILD FLAGS
012FC 3605F F0004   8720              SETQBIT 4                         ;SET ILD RUNNING
012FD 321FF F00BA   8721              QLDRAM ILDFLG
012FE 373FD F00B4   8722              RDRAMB R13,ILDDRV                 ;GET ILD DRIVE NO.
012FF 433F7 FFFFF   8724              RESETR   R7                       ;$
01300 3535F F2000   8733              SETRBIT R15,2000                  ;SET THE PULLING FLAG
                    8734              JSR.P SEL01                       ;SELECT THE DRIVE
01301 E0401 1F187   8734 +            SETPGE SEL01_SHR_H#B & JSRP SEL01
                    8735              JSR.P RELDRIV                     ;RELEASE & DE-SELECT THE DRIVE
01302 E0401 1F60F   8735 +            SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
01303 FFFFF 1F26B   8736              JSR RST1BIT                       ;RESET ILD BIT FOR THIS DRIVE
01304 333F7 FFFFF   8737              OR ZB,B,,R7
01305 FFFFF A37FF   8738              CRTS ALUZ                         ;IF DRIVE SELECTED, NORMAL RETURN
01306 371FE F02FF   8739    RTRN5:    MOVE RAME,IREG & NOP RETURN5
01307 371F9 F02FE   8740              MOVE RAME,IREGH & NOP RETURN5A
01308 FFFFF 6F7FF   8741              JIR                               ;ELSE, TAKE ERROR RETURN
                    8742
                    8743
                    8744    TITLE FORMAT/VERIFY ILD'S
                    8745
                    8746    FORMAT:   EQU H#38          ;FORMAT JUMP LOC.
                    8750
                    8751    ILD100:   EQU $
01309 373F2 F00B8   8756              RDRAMB R2,STCYL                   ;GET LAST STARTING CYL. AND...
0130A 373F1 F00B9   8757              RDRAMB R1,NDCYL                   ;ENDING CYL. FOR CHECKING
                    8758    ;
0130B 4505E F0020   8759              TSTRBIT R14,20                    ;SPECIAL FLAG SET?
0130C 603EE 33B15   8760              XOR AQ,B,R14,R14 & BNZ ILD102     ;IF SET, RESET IT & JUMP
0130D 0F322 E12CA   8764              CALL R2,CKFLAG                    ;GET DRIVE FLAGS
0130E FFFFF 33346   8765              BOZ ILD112                        ;IF NO FLAG BITS, JUMP
0130F 373F2 F00B8   8770              RDRAMB R2,STCYL                   ;GET START CYLINDER AND...
01310 373F1 F00B9   8771              RDRAMB R1,NDCYL                   ;ENDING CYLINDER FOR CHECKING
```

Addr Line FORMAT/VERIFY ILD'S

```
                    8772    ;
01311 2D1F2 F020A   8773            SUBB DB,FOUTL,RAME,R2 & NOP MAXCYLI ;IF STARTING CYL. GREATER THAN...
01312 433F0 3532A   8774            RESETR R0 & CJMP ALB,ILD104         ;MAX., CHECK FURTHER
01313 2D1F1 F020A   8775            SUBB DB,FOUTL,RAME,R1 & NOP MAXCYLI ;IF ENDING CYL. GREATER THAN...
01314 FFFFF 3532A   8776            CJMP ALB,ILD104                     ;MAX., CHECK FURTHER
                    8777    ;
01315 29112 FFFFF   8778    ILD102: SUBB AB,FOUTL,R1,R2                 ;IF STARTING CYL. IS > THAN...
01316 433F0 3532C   8779            RESETR R0 & CJMP ALB,ILD105         ;END CYL., GET NEW CYL. NUMBERS
                    8780    ;
                    8781    ;   Display Starting and Ending cylinders
                    8782    ;
01317 0F322 E1170   8783            CALL R2,STRTCYL                     ;DISPLAY STARTING CYLINDER
01318 0F322 E1177   8784            CALL R2,ENDCYL                      ;DISPLAY ENDING CYLINDER
01319 0F322 E125B   8785            CALL R2,YESNO                       ;ARE THESE NUMBERS OK?
0131A FFFFF 3332C   8786            BOZ ILD105                          ;IF 'NO' GET NEW NUMBERS
                    8787    ;
0131B 370FF F00B8   8788    ILD102A: RDRAMQ STCYL
0131C 661F0 F00B9   8789            XOR DQ,FOUTL,RAME,ZERO & NOP NDCYL ;IF THE START CYL. IS NOT EQUAL...
0131D FFFFF 33B21   8790            BNZ ILD103                          ;TO THE END CYL., DISPLAY HEAD NOS.
0131E 370FF F00BB   8791            RDRAMQ STHD
0131F 2E1F0 F00BC   8792            SUBB DQ,FOUTL,RAME,ZERO & NOP NDHD ;IF THE START HEAD IS GREATER...
01320 FFFFF 35338   8793            CJMP ALB,ILD110                     ;THAN END HEAD, GET NEW HEAD NOS.
                    8794    ;
01321 373F1 F00BD   8795    ILD103: RDRAMB R1,ILDNBR                    ;GET ILD NUMBER
01322 65051 F0002   8796            TSTRMSK R1,2                        ;CHECK FOR ILD 2
01323 FFFFF 33B27   8797            BNZ ILD103A                         ;IF NOT, JUMP
01324 370FF F00BC   8798            RDRAMQ NDHD                         ;GET ENDING HEAD NO.
01325 2E1F0 F00BB   8799            SUBB DQ,FOUTL,RAME,ZERO & NOP STHD ;IF START HEAD IS GREATER THAN...
01326 FFFFF 34338   8800            CJMP AGB,ILD110                     ;ENDING HEAD, GET NEW HEAD NOS.
                    8801    ;
                    8802    ;   Display Starting and Ending Heads
                    8803    ;
01327 0F322 E118D   8804    ILD103A: CALL R2,STRTHD                     ;DISPLAY STARTING HEAD NO.
01328 0F322 E119A   8805            CALL R2,ENDHD                       ;DISPLAY ENDING HEAD NO.
01329 FFFFF E1090   8806            JMP ILD006                          ;ARE THESE NUMBERS OK?
                    8807    ;
0132A 2D1F1 F020A   8808    ILD104: SUBB DB,FOUTL,RAME,R1 & NOP MAXCYLI ;END CYL.> MAX. USABLE CYL?
0132B 433F0 35346   8809            RESETR R0 & CJMP ALB,ILD112         ;IF YES, SET MAX. LIMITS
                    8810    ;
                    8811    ;   Get new Start & End cylinders.
                    8812    ;
0132C 0F322 E11A9   8813    ILD105: CALL R2,GTSTCYL                     ;GET STARTING CYL. FROM OPERATOR
0132D 433F0 FFFFF   8814            RESETR R0
0132E 0F322 E11BA   8815            CALL R2,GTNDCYL                     ;GET END CYL. FROM OPERATOR
                    8816    ;
0132F 2D1F0 F020A   8817            SUBB DB,FOUTL,RAME,R0 & NOP MAXCYLI ;IF ENDING CYL.> MAX...
01330 FFFFF 35334   8818            CJMP ALB,ILD108                     ;CHECK FURTHER
                    8819    ;
01331 2D1F0 F00B8   8820    ILD107: SUBB DB,FOUTL,RAME,R0 & NOP STCYL  ;IF STARTING CYLINDER IS GREATER...
01332 FFFFF 341F6   8821            CJMP AGB,BADCYL                     ;THAN ENDING CYL., GET NEW ENTRY
01333 FFFFF E131B   8822            JMP ILD102A                         ;DISPLAY START & END HEADS
                    8823    ;
01334 373F1 F00BD   8824    ILD108: RDRAMB R1,ILDNBR                    ;GET ILD NUMBER
01335 65051 F0002   8825            TSTRMSK R1,2                        ;CHECK FOR VERIFY
```

Addr Line FORMAT/VERIFY ILD'S

```
01336 FFFFF 331F6  8829            BOZ BADCYL                      ;IF VERIFY, BAD END CYL. ENTRY
01337 FFFFF E1B6E  8831            JMP ILD109
                   8832    ;
                   8833    ;   Get new Start & End heads
                   8834    ;
01338 0F322 E11C7  8835    ILD110: CALL R2,GTSTHED                 ;GET START HEAD FROM OPERATOR
01339 0F322 E11E6  8836            CALL R2,GTNDHED                 ;GET ENDING HEAD FROM OPERATOR
                   8837    ;
0133A 370FF F00BD  8838            RDRAMQ ILDNBR                   ;GET ILD NUMBER
0133B 6605F F0002  8839            TSTQMSK 2                       ;CHECK FOR ILD 2
0133C FFFFF 33B40  8840            BNZ $+4                         ;IF NOT, JUMP
0133D 273E2 FFFFF  8841            SAB DZ,B,IREG,R2
0133E 2D1F0 F00BB  8842            SUBB DB,FOUTL,RAME,R0 & NOP STHD ;IF START HEAD > END...
0133F FFFFF 341E3  8843            CJMP AGB,STHED3                 ;HEAD, GET NEW HEAD NOS.
                   8844    ;
01340 370FF F00B8  8845            RDRAMQ STCYL
01341 661F0 F00B9  8846            XOR DQ,FOUTL,RAME,ZERO & NOP NDCYL ;IF START CYL. NOT = TO END...
01342 FFFFF 33892  8847            BNZ ILD06A                      ;CYL., START FORMAT OR VERIFY
                   8848    ;
01343 2D1F0 F00BB  8849            SUBB DB,FOUTL,RAME,R0 & NOP STHD ;IF STARTING HEAD IS GREATER...
01344 FFFFF 341F4  8850            CJMP AGB,NDHED3                 ;THAN END HEAD, GET NEW END HEAD
01345 FFFFF E1092  8851            JMP ILD06A                      ;START FORMAT OR VERIFY
                   8852    ;
                   8853    ;   SET UP MAX. CYLINDER & HEAD LIMITS
                   8854    ;
01346 3710F F00B5  8855    ILD112: RESETRAM CURCYL                 ;RESET CURRENT CYLINDER,
01347 3710F F00B8  8856            RESETRAM STCYL                  ;STARTING CYL. AND...
01348 3710F F00BB  8857            RESETRAM STHD                   ;STARTING HEAD
                   8858    ;
01349 0F322 E12CA  8875            CALL R2,CKFLAG                  ;GET DRIVE FLAGS
0134A FFFFF 333DB  8876            BOZ ILD150                      ;IF NO FLAG BITS, JUMP
0134B 373F0 F020A  8882            RDRAMB R0,MAXCYL1               ;MAX. CYLINDER
0134C 373F1 F020D  8883    ILD112B: RDRAMB R1,MAXHD1               ;MAX. HEAD
0134D 3410F F00B9  8884    ILD113: ALDRAM R0,NDCYL                 ;SAVE ENDING CYLINDER
0134E 3411F F00BC  8885            ALDRAM R1,NDHD                  ;SAVE ENDING HEAD
0134F 4505B F0030  8886            TSTRBIT R0,30                   ;ANY RECORD LENGTH BITS?
01350 FFFFF 33353  8887            BOZ ILD115                      ;IF NOT, MUST FORMAT ENTIRE PACK
01351 3535E F0020  8888            SETRBIT R14,20                  ;SET SPECIAL FLAG
01352 FFFFF E1309  8889            JMP ILD100
                   8890
                   8891
                   8892    TITLE ILD 1 - FORMAT THE PACK
```

Addr Line ILD 1 - FORMAT THE PACK

```
                    8894
01353 FFFFF 1F18B   8896    ILD115:   JSR GETFLAG$                    ;$GET HEADER FLAGS THIS DRIVE
01354 55358 F0020   8897              RSTRBIT R8,20                   ;$RESET 448 PREP. FACTOR
01355 35358 F0010   8898              SETRBIT R8,10                   ;$SET 116-WORD PREP.
01356 3418D FFFFF   8899              ALDREG R8,RAM1                  ;$SAVE FLAGS THIS DRIVE
                    9026    ;
                    9027    ILD115J:  EQU $
01357 37350 F0014   9033              RDMSKB R0,D#20
01358 0F322 E0E78   9034              CALL R2,ISSCMDA1                ;MSG: "DO YOU WANT TO FORMAT DR. "
01359 0F322 E126E   9035              CALL R2,PRNTDRV                 ;PRINT DRIVE NO.
0135A 37350 F0045   9036              RDMSKB R0,D#69                  ;MSG: "?"
0135B 0F322 E0E7B   9037              CALL R2,ISSCMDA1
                    9038    ;
0135C 0F322 E1261   9039              CALL R2,YESNOA                  ;WAIT FOR 'YES' OR 'NO'
0135D E100B 33133   9041              CTL HSSM,RSTUSEL & BOZ ILD28    ;$DE-SELECT DRIVE; 'NO', TERMINATE
                    9045    ;
                    9046    ;     START FORMATTING
                    9047    ;
0135E 3710F F01EF   9048              RESETRAM MSGFLG                 ;RESET LOOP & MESSAGE FLAGS
0135F 433F5 FFFFF   9050              RESETR   R5                     ;$Reset flag reg.
01360 FFFFF 1F273   9055              JSR      FRMTDAT                ;GET THE FORMAT DATA
01361 37352 F1364   9056    ILD116:   RDMSKB R2,$+3                   ;SAVE ILD CONTINUATION LOC...
01362 FFFFF 1F183   9057              JSR SAVCONT                     ;IN CASE OF SELECTION ERROR
01363 0F322 E1239   9058              CALL R2,ILDRUN                  ;SELECT ILD DRIVE
                    9062    ;
01364 3535F F0008   9063              SETRBIT R15,8                   ;SET RTZ FLAG
01365 37352 F1368   9064              RDMSKB R2,ILD120
01366 FFFFF 1F183   9065              JSR SAVCONT                     ;SAVE ILD CONTINUATION LOC.
01367 3710F F0018   9066              RESETRAM SEN415                 ;RESET SM-RETRY COUNT
                    9067    ;
01368 4505F F0100   9068    ILD120:   TSTRBIT R15,100                 ;CHECK FOR WRITE PROTECT
01369 0F322 33AA5   9069              ADD DZC,B,SE0AR,R2 & BNZ WRTPROT ;IF SET, WRITE PROTECT ROUTINE
                    9070    ;
0136A 4505F F0400   9071              TSTRBIT R15,400                 ;CHECK SM-RETRY FLAG
0136B FFFFF 33B72   9072              BNZ ILD121                      ;IF SET, JUMP
0136C 4505F F0008   9073              TSTRBIT R15,8                   ;RTZ FLAG SET?
                    9074              BNZ.P ILD1RTZ                   ;IF YES, RE-ZERO THE DRIVE
0136D E0403 33B3B   9074 +            SETPGE ILD1RTZ_SHR_H#B & BNZP ILD1RTZ
0136E 5535F F0408   9075    ILD120A:  RSTRBIT R15,408                 ;RESET SM-RETRY/RTZ FLAGS
0136F 0F322 E115A   9076              CALL R2,ILDSK0                  ;POSITION THE DRIVE
```

```
 Addr            Line ILD 1 -- FORMAT THE PACK

01370 5535F F0408  9077            RSTRBIT R15,408                 ;RESET THE FLAGS AGAIN
01371 FFFFF E1374  9078            JMP $+3
                   9079    ;
                   9080    ILD121:  MOVSMB   CURHD,MAIN1LD3        ;Display current Head no.
01372 373F0 F00B6  9080 +          RDRAMB   R0,CURHD
01373 3410F F00B3  9080 +          ALDRAM   R0,MAIN1LD3
01374 FFFFF 1F1A7  9081            JSR CHKMSG1                     ;CHECK MSG. SENT 1
01375 FFFFF 33B7F  9082            BNZ ILD122                      ;IF SENT, SKIP MESSAGE
                   9083    ;
01376 37350 F0015  9084    ILD121A: RDMSKB R0,D#21
01377 0F322 E0E78  9085            CALL R2,ISSCMDA1                ;MSG: "FORMATTING DR. "
01378 0F322 E126E  9086            CALL R2,PRNTDRV                 ;PRINT DRIVE NO.
01379 37350 F001C  9087            RDMSKB R0,D#28
0137A 0F322 E0E7B  9088            CALL R2,ISSCMDA1                ;MSG: ", CYL. "
0137B FFFFF 1F22F  9089            JSR RECORD1                     ;GET CUR. HEAD
0137C 373FB F00B5  9090            RDRAMB R11,CURCYL               ;GET CUR. CYLINDER
0137D 0F322 E1263  9091            CALL R2,DSPLY                   ;DISPLAY & PRINT CUR. CYLINDER
0137E FFFFF 1F1A4  9092            JSR SETMSG1                     ;SET MSG. SENT ) FLAG
                   9097    ;
                   9098    ILD122:  EQU $
                   9100             JSR.P    FORMDAT                ;$Load format data to MSSM
0137F E0403 1F759  9100 +          SETPGE FORMDAT_SHR_M#B & JSRP FORMDAT
01380 0F322 E1239  9105            CALL R2,ILDRUN                  ;SELECT ILD DRIVE IF NECESSARY
01381 37352 F1383  9134            RDMSKB R2,ILD123                ;RETURN LOC. FROM CONV100
01382 FFFFF E11FB  9135            JMP CONV100                     ;SET UP SENSE BYTES
                   9136    ;
01383 FFFFF 1F18B  9137    ILD123:  JSR GETFLAG$                    ;GET HEADER FLAGS THIS DRIVE
01384 4505F F0010  9138            TSTRBIT R15,10                  ;CHECK DEFECTIVE FLAG
01385 FFFFF 3338A  9139            BOZ ILD125                      ;IF NOT SET, NORMAL FORMAT
                   9140    ;
                   9141    ;    FORMAT DEFECTIVE TRACK
                   9142    ;
01386 3535B F0001  9143            SETRBIT R8,1                    ;SET DEFECTIVE FLAG
01387 373FB F00B9  9144            RDRAMB R11,NDCYL                ;GET ALTERNATE (ENDING) CYL.
01388 373FC F00BC  9145            RDRAMB R12,NDHD                 ;GET ALTERNATE (ENDING) HEAD
                   9146    ;
01389 FFFFF 1F230  9147            JSR RECORD1+1                   ;SAVE HEAD & SET REC.= 1
0138A 37359 F0001  9148    ILD125:  RDMSKB   R9,1                   ;Starting record is always 1
0138B 343B3 FFFFF  9149            RTOR R11,R3                     ;MOVE CYL. TO R3
                   9155            JSR.P DRVHDR                    ;LOAD THE HEADER INFO.
0138C E0403 1F569  9155 +          SETPGE DRVHDR_SHR_M#B & JSRP DRVHDR
```

Addr Line ILD 1 - FORMAT THE PACK

```
0138D 9198F FFFFF  9161            WRTL HSSM,LDDRFLG,R8              ;LOAD HEADER FLAGS
0138E 373F0 F02D6  9163            RDRAMB R0,AMPREC                  ;GET MAX. RECORD
0138F 9160F FFFFF  9164            WRTL HSSM,LDUBCL,R0               ;LOAD NO. OF RECORDS TO FORMAT
01390 373F0 F02CD  9165            RDRAMB R0,RECLNTH                 ;GET THE RECORD LENGTH,
                   9166            JSR.P FOUR5DN                     ;CONVERT IT TO BYTES AND...
01391 E0400 1F368  9166 +          SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
01392 083F0 FFFFF  9167            INCRB R0                          ;INCR. BYTE COUNT BY 1
01393 9120F FFFFF  9168            WRTL HSSM,LDSBCL,R0               ;LOAD THE 8 LSB'S
01394 A131F FFFFF  9169            WRTU HSSM,LDSBCM,R1               ;LOAD THE 8 MSB'S
01395 B1100 F0000  9170            WRTE HSSM,LDSARM & NOP 0          ;STARTING ADDRESS...
01396 B1000 F0000  9171            WRTE HSSM,LDSARL & NOP 0          ;IS ALWAYS 0
                   9172    ;
01397 B1F00 F0038  9173            WRTE HSSM,LDXMCJ & NOP FORMAT     ;INITIATE FORMAT OPERATION
01398 E100D FFFFF  9179            CTL HSSM,XMCJCMD
01399 E1005 FFFFF  9180            CTL HSSM,WRTCONT                  ;SET WRITE CONTINUE FLAG
0139A 37355 F3800  9181            RDMSKB R5,3800                    ;SET WRITE & FORMAT ACTIVE FLAGS
0139B 37350 F13BC  9182            RDMSKB R0,ILD129                  ;SAVE SM-ERROR RETURN...
                   9183            JSR.P SAVSMRTN                    ;IN CASE OF AN ERROR
0139C E0400 1F4C6  9183 +          SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
0139D 37352 F1372  9184            RDMSKB R2,ILD121                  ;SAVE RETURN LOC. FOR NEXT...
0139E FFFFF 1F183  9185            JSR SAVCONT                       ;TRACK OR LOOP-ON-ERROR
0139F 37350 F8647  9186            RDMSKB R0,8647                    ;GET 55-MS. TIMER...
                   9187            JSR.P SAVSMTIM                    ;AND SAVE IT
013A0 E0400 1F4C2  9187 +          SETPGE SAVSMTIM_SHR_H#B & JSRP SAVSMTIM
013A1 37350 F13A8  9188            RDMSKB R0,ILD128                  ;FORMAT SM-DONE RETURN LOC.
                   9189            JSR.P SAVIRPT                     ;SAVE IT
013A2 E0400 1F4C0  9189 +          SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
                   9190    ;
013A3 4505F F0400  9191            TSTRBIT R15,400                   ;SM-RETRY FLAG?
                   9192            CJSR.P ALUZ,SMBOOK                ;IF NOT, DO BOOKKEEPING
013A4 E0401 130D0  9192 +          SETPGE SMBOOK_SHR_H#B & CJSRP ALUZ,SMBOOK
                   9193    ;
                   9194    ILD127: JSR.P SMIRPT                      ;CHECK FOR AN INTERRUPT
013A5 E0400 1F37C  9194 +          SETPGE SMIRPT_SHR_H#B & JSRP SMIRPT
                   9195    ;
                   9196            JSR.P SMTIMER                     ;THERE WAS NONE, DECR. TIMER &...
013A6 E0400 1F4DB  9196 +          SETPGE SMTIMER_SHR_H#B & JSRP SMTIMER
013A7 FFFFF E13A5  9197            JMP ILD127                        ;CHECK AGAIN
                   9198    ;
                   9199    ; ONE TRACK DONE.
                   9200    ;
013A8 E1000 FFFFF  9201    ILD128: CTL HSSM,DIFRST                   ;DISK INTERFACE RESET
013A9 B1F00 F0002  9202            WRTE HSSM,LDXMCJ & NOP HSWAIT     ;MPC WAIT LOOP
013AA E100D 1F280  9203            CTL HSSM,XMCJCMD & JSR ILDCYHD    ;DISPLAY CUR. CYL. & HEAD
                   9204    ;
```

```
 Addr            Line ILD 1 - FORMAT THE PACK

013AB 4505F F0010  9205            TSTRBIT R15,10                  ;CHECK DEFECTIVE FLAG
013AC 603FF 33BC2  9206            XOR A0,B,R15,R15 & BNZ ILD132   ;IF SET, RESET IT & SEEK TO ALT.
013AD 4505E F0200  9207            TSTRBIT R14,200                 ;CHECK FOR ALTERNATE FLAG
013AE 603EE 33BCB  9208            XOR A0,B,R14,R14 & BNZ ILD133   ;IF SET, RESET IT & START VERIFY
                   9209    ;
013AF FFFFF 1F2EB  9210            JSR LIMIT                       ;CHECK IF DONE
013B0 FFFFF 33BCB  9211            BNZ ILD133                      ;IF DONE, START VERIFY
                   9212    ;
013B1 5535F F0400  9213            RSTRBIT R15,400                 ;RESET SM-RETRY FLAG
                   9214            TSTMBIT MSG2,MSGFLG             ;Message 2 flag set?
013B2 373F0 F01EF  9214 +          RDRAMB   R0,MSGFLG
013B3 45150 F0004  9214 +          TSTBIT   R0,MSG2
013B4 FFFFF 333B7  9215            BOZ $+3                         ;IF NOT, JUMP
013B5 55350 F0006  9216            RSTRBIT R0,MSG2+MSG1            ;RESET MESSAGE 1 & 2 FLAGS
013B6 3410F F01EF  9217            ALDRAM R0,MSGFLG
013B7 4505F F1000  9218            TSTRBIT R15,1000                ;CHECK ADV. CYLINDER FLAG
013B8 603FF 33BBE  9219            XOR A0,B,R15,R15 & BNZ ILD130   ;IF SET, RESET IT & JUMP
013B9 5535F F0800  9220            RSTRBIT R15,800                 ;RESET ADV. HEAD FLAG
                   9221            JSR.P ADVHEAD                   ;SET NEXT HEAD
013BA E0400 1F51C  9221 +          SETPGE ADVHEAD_SHR_H#B & JSRP ADVHEAD
013BB FFFFF E10D2  9222            JMP ILD20                       ;GO THROUGH WAIT LOOP
                   9223
                   9224    ;   FORMAT ERROR RETRY
                   9225    ;
                   9226    ILD129:  JSR.P WAITCON                  ;WAIT TO CONTINUE ROUTINE
013BC E0400 1F521  9226 +          SETPGE WAITCON_SHR_H#B & JSRP WAITCON
013BD FFFFF E1383  9227            JMP ILD123
```

Addr Line ILD 1 - FORMAT THE PACK

```
                    9229
                    9230    ;   FORMAT ADVANCE CYLINDER
                    9231    ;
013BE 341BF F01FE   9232    ILD130:   ALDRAM R11,SAVCYL                 ;SAVE NEXT CYLINDER
013BF 0F322 E1263   9233              CALL R2,DSPLY                     ;DISPLAY & PRINT NEXT CYLINDER
013C0 0F322 E115A   9239              CALL R2,ILDSK0                    ;SEEK TO NEXT CYLINDER
013C1 FFFFF E10D6   9240              JMP ILD21                         ;CONTINUE FORMAT
                    9241    ;
013C2 373F3 F00BA   9242    ILD132:   RDRAMB R3,ILDFLG                  ;GET ILD FLAGS
013C3 45053 F0200   9243              TSTRBIT R3,200                    ;FORMAT DEFECTIVE ALTERNATE?
                    9244              BNZ.P ILD410                      ;YES, JUMP
013C4 E0403 33CE8   9244 +            SETPGE ILD410_SHR_N#B & BNZP ILD410
                    9245    ;
013C5 3535F F0400   9246              SETRBIT R15,400                   ;SET SM-RETRY FLAG
013C6 3419F F023A   9247              ALDRAM R9,SAVREC                  ;SAVE RECORD BEFORE SEEKING
013C7 0F322 E115A   9248              CALL R2,ILDSK0                    ;SEEK TO ALTERNATE TRACK
                    9249    ;
013C8 FFFFF 1F22B   9250              JSR STRTOVR                       ;GET STARTING CYL. & HEAD
013C9 3535E F0200   9251              SETRBIT R14,200                   ;SET ALTERNATE FLAG AGAIN
013CA FFFFF E137F   9252              JMP ILD122                        ;FORMAT THE ALTERNATE TRACK
                    9253    ;
013CB 5535F F1C00   9254    ILD133:   RSTRBIT R15,1C00                  ;RESET SOME FLAGS
013CC FFFFF 1F22B   9255              JSR STRTOVR                       ;GET STARTING CYL. & HEAD
013CD 3705F F0002   9256              RDMSKQ 2
013CE 321FF F00BD   9257              QLDRAM ILDNBR                     ;SET ILD NO. 2 (VERIFY)
013CF 373F3 F00BA   9258              RDRAMB R3,ILDFLG                  ;GET ILD FLAGS
013D0 3710F F01EF   9263              RESETRAM MSGFLG                   ;RESET LOOP/MSG. SENT FLAGS
                    9268    ;
013D1 650F0 F00B9   9269              XOR DB,Q,RAME,R0 & NOP NDCYL      ;IF START CYL.= END...
013D2 FFFFF 333D5   9270              BOZ ILD134                        ;CYL., START VERIFY
                    9271    ;
013D3 370FF F02D5   9272              RDRAMQ MAXHDA                     ;GET MAX. HEAD
013D4 321FF F00BC   9273    ILD133A:  QLDRAM NDHD                       ;SET END HEAD = MAX. HEAD
                    9274    ;
013D5 35353 F0010   9275    ILD134:   SETRBIT R3,10                     ;SET FORMAT-TO-VERIFY FLAG
013D6 3413F F00BA   9276              ALDRAM R3,ILDFLG
013D7 FFFFF 1F233   9277              JSR ILDDSP                        ;DISPLAY ILD NO. 2 & DRIVE NO.
013D8 FFFFF E1439   9282              JMP ILD200                        ;START VERIFY
```

Addr Line ILD 1 - FORMAT THE PACK

```
                 9285   ;
                 9286   ;    AN SM-INTERRUPT HAS EXPECTED, BUT NONE HAS RECEIVED.
                 9287   ;
013D9 37350 F001C 9288   ILD140:   RDMSKB R0,1C                    ;SET SENSE BYTE 7
013DA E1000 E0AFB 9289             CTL HSSM,DIFRST & JMP SEL24     ;SEND EQUIP. CHECK
                 9310
                 9311
                 9312   ;    WE HAVE TRIED TO DETERMINE THE DRIVE TYPE, BUT WERE UNABLE TO READ A
                 9313   ;FLAG BYTE FROM THE HEADER.  THIS ROUTINE WILL ISSUE A SEEK TO THE MAX.
                 9314   ;CYLINDER OF EACH DRIVE TYPE UNTIL ONE IS FOUND THAT DOES NOT GENERATE A
                 9315   ;"SEEK ERROR" STATUS.  WHEN THE DRIVE TYPE IS FOUND THE VALUE WILL BE SAVED
                 9316   ;IN THE FLAG BYTE FOR THIS DRIVE.  ALSO, THE MAX. USEABLE CYLINDER AND MAX.
                 9317   ;DRIVE CYLINDER WILL BE SAVED.  A RE-ZERO WILL BE ISSUED TO THE DRIVE
                 9318   ;WHETHER OR NOT A "SEEK ERROR" OCCURRED.
                 9319   ;
013DB 3710F F01FD 9320   ILD150:   RESETRAM SEEDWRD
013DC 3710F F00B6 9321             RESETRAM CURHD                  ;RESET CURRENT HEAD
013DD 3710F F024E 9322             RESETRAM CACHECNT               ;RESET RETRY COUNT
                 9323   ;
013DE 370FF F01FD 9324   ILD150A:  RDRAMQ SEEDWRD                  ;GET THE SEED WORD
013DF 0615E F13E2 9325             ADD DQ,FOUTL,MSK2,IREG & NOP $+D#3 ;COMPUTE THE JUMP LOC...
013E0 37929 5F3FD 9326             OR DZ,FOUTU,SEQAR,IREGH & LDAR ILD159 ;ACCORDING TO THE SEED WORD
013E1 0A1FF 6F1FD 9327             ADD ZQC,FOUTL,,RAME & JIR SEEDWRD  ;INCR. SEED WORD & JUMP
                 9328   ;
013E2 FFFFF E13EF 9329             JMP ILD151
013E3 FFFFF E13F9 9330             JMP ILD156
013E4 FFFFF E13F1 9331             JMP ILD152
013E5 FFFFF E13F3 9332             JMP ILD153
013E6 FFFFF E13F5 9333             JMP ILD154
013E7 FFFFF E13FB 9334             JMP ILD158
013E8 FFFFF E13FB 9335             JMP ILD158
013E9 FFFFF E13F7 9336             JMP ILD155
                 9337   ;
013EA 37350 F0039 9338   ILD150B:  RDMSKB R0,D#57                  ;MSG: "SEEK ERROR "
013EB 0F322 E0E78 9339             CALL R2,(SSCMDA1
013EC 0F322 E126E 9340             CALL R2,PRNTDRV                 ;PRINT DRIVE NUMBER
013ED 37350 F0024 9341             RDMSKB R0,D#36
013EE FFFFF E1134 9342             JMP ILD28+1                     ;TERMINATE THE ILD
                 9343   ;
013EF 37352 F03FF 9344   ILD151:   RDMSKB R2,D#1023                ;SEEK CYL.= 1023
013F0 37343 5F801 9345             RDREGB MSK1,R3 & JAR H#1+DFECT  ;DRIVE TYPE = 1 (CDC 1.2 GB.)
                 9346   ;
013F1 37352 F034B 9347   ILD152:   RDMSKB R2,MAXCYL2               ;SEEK CYL.= 843
013F2 37343 5F802 9351             RDREGB MSK1,R3 & JAR H#2        ;DRIVE TYPE = 2 (CDS513)
                 9353   ;
013F3 37352 F0336 9354   ILD153:   RDMSKB R2,MAXCYL3/2             ;SEEK CYL.= 822
013F4 37343 5F803 9355             RDREGB MSK1,R3 & JAR H#3        ;DRIVE TYPE = 3 (FUJI337)
                 9356   ;
013F5 37352 F026F 9357   ILD154:   RDMSKB R2,MAXCYL5               ;SEEK CYL.= 623
```

Addr Line ILD 1 - FORMAT THE PACK

```
013F6 37343 5F804  9359          RDREGB MSK1,R3 & JAR H#4          ;TYPE = 4 (SWALLOW - HARD SECT.)
                   9367    ;
013F7 37352 F03FF  9368    ILD155:  RDMSKB R2,D#1023               ;SEEK CYL.=
013F8 37343 5F805  9369          RDREGB MSK1,R3 & JAR H#5          ;DRIVE TYPE = 5 (Swal-4)
                   9370    ;
013F9 37352 F03FF  9371    ILD156:  RDMSKB R2,D#1023               ;SEEK CYL.= 1023 (CDC-368)
013FA 37343 5F806  9379          RDREGB MSK1,R3 & JAR H#6          ;DRIVE TYPE = 6 (CDC-368)
                   9380    ;
013FB 37352 F03FF  9381    ILD158:  RDMSKB R2,D#1023               ;SEEK CYL.=
013FC 37343 5F800  9382          RDREGB MSK1,R3 & JAR H#0          ;DRIVE TYPE = 0
                   9383    ;
013FD 45052 F0C00  9384    ILD159:  TSTRBIT R2,H#C00               ;ISOLATE CYL. MSB'S,
013FE 325F0 4F001  9385          OR ZQ,RSBZ,,R0 & PSHLD 1
013FF 335F0 887FF  9386          RSHFTZ R0                         ;MOVE THEM INTO POSITION...
01400 9220F FFFFF  9387          WRTL HSSMA,LDDRBO,R0              ;AND LOAD THEM
01401 60324 4F007  9388          XOR AQ,B,R2,R4 & PSHLD 7          ;CYLINDER LSB'S TO R4
01402 E1009 887FF  9389          CTL HSSM,SETAG2 & CLPCNT EVCNTZ   ;ISSUE 'SET HEAD'
01403 E1013 FFFFF  9390          CTL HSSM,RSTAG2
                   9391    ;
01404 37351 F003D  9392          RDMSKD R1,3D
01405 D0FF1 FFFFF  9393          RDXFR SMC,,R1                     ;SET SEEK TIME-OUT TIMER
01406 9224F C0007  9394          WRTL HSSMA,LDDRBO,R4 & LDCT 7     ;LOAD SEEK CYLINDER
01407 3412F F00B5  9395          ALDRAM R2,CURCYL                  ;SAVE SEEK CYLINDER
01408 E1008 98408  9396          CTL HSSM,SETAG1 & CJPCNT EVCNTZ,$ ;START THE SEEK
01409 E1012 4F028  9397          CTL HSSM,RSTAG1 & PSHLD 40
0140A C1040 887FF  9398          RD HSSM,RDDRST1,R0 & CLPCNT EVCNTZ ;DELAY 8-µsec FOR STATUS
0140B C1040 5F40B  9399          RD HSSM,RDDRST1,R0 & LDAR $       ;READ DRIVE STATUS
0140C 45050 F008E  9400          TSTRBIT R0,SEEKERR+FAULT+ONCYL+SKEND ;ANY GOOD OR BAD STATUS?
0140D 133F1 33C0F  9401          DECRB R1 & BNZ $+2                ;IF YES, JUMP; DECR. TIMER
0140E C1040 73430  9402          RD HSSM,RDDRST1,R0 & CJARB ALUZ,ILD160 ;IF NO TIME-OUT, JUMP $-3
                   9403    ;
0140F FFFFF 33430  9404          BOZ ILD160                        ;IF TIME-OUT, GIVE UP
01410 45150 F000C  9405          TSTBIT R0,SEEKERR+FAULT           ;SEEK ERROR OR FAULT?
01411 FFFFF 33C30  9406          BNZ ILD160                        ;IF EITHER, JUMP
01412 6605F F0082  9407          TSTQMSK ONCYL+SKEND               ;SEEK END & ON CYLINDER?
01413 C1040 73414  9408          RD HSSM,RDDRST1,R0 & CJARB ALUZ,$+1 ;YES, FALL THRU; NO, JUMP BACK
                   9409    ;
01414 3705F F003C  9410          RDMSKQ 3C
01415 D0FF0 FFFFF  9411          RDXFR SMC,,R0
01416 35350 F0F06  9412          OR DB,B,MSK2,R0 & NOP H#F06       ;SET 200-µsec. INTO R0
01417 29110 FFFFF  9413          SUBB AB,FOUTL,R1,R0               ;IF THE SEEK TOOK MORE THAN...
01418 FFFFF 3541F  9414          CJMP ALB,ILD159B                  ;200 µsec., GOOD SEEK
                   9415    ;
01419 370FF F024E  9416          RDRAMQ CACHECNT                   ;GET THE RETRY COUNT
0141A FFFFF B3BEA  9417          CJPOP NALUZ,ILD150B               ;IF NOT 0, ERROR
0141B 0A1FF F024E  9418          ADD ZQC,FOUTL,,RAME & NOP CACHECNT ;INCR. RETRY COUNT
0141C 370FF F01FD  9419          RDRAMQ SEEDWRD
0141D 121FF F01FD  9420          SBA ZQ,FOUTL,,RAME & NOP SEEDWRD  ;DECR. THE SEED WORD AND...
0141E FFFFF E1431  9421          JMP ILD160+1                      ;TRY THIS DRIVE AGAIN
                   9422    ;
```

```
 Addr            Line ILD 1 - FORMAT THE PACK

0141F FFFFF 5F3FD  9423   ILD159B:  LDAR ILD159
01420 65052 F03FF  9424             TSTRMSK R2,D#1023               ;CUR. CYL.= 1023?
01421 34731 33C28  9425             OR AZ,LSBZ,R3,R1 & BNZ ILD159C  ;NO, JUMP; SHIFT DRIVE TYPE
01422 65053 F0006  9431             TSTRMSK R3,6                    ;DRIVE TYPE 6 (CDC-368)?
01423 FFFFF 33C25  9432             BNZ $+2                         ;NO, JUMP
01424 37342 5FCC0  9433             RDREGB MSK1,R2 & JAR D#1216     ;CDC-368 MAX. CYL.= 1216
01425 65053 F0001  9434             TSTRMSK R3,1+DFECT              ;DRIVE TYPE 1 (CDC 1.2 GB.)?
01426 FFFFF 33C30  9435             BNZ ILD160                      ;NO, JUMP
01427 37342 5FE61  9436             RDREGB MSK1,R2 & JAR MAXCYL1    ;CDC 1.2 MAX. CYL.= 1633
                   9437   ;
01428 65053 F0004  9438   ILD159C:  TSTRMSK R3,4                    ;DRIVE TYPE 4 (SWALLOW-5)?
01429 C3010 33C2D  9439             RD SCP,PRGTGL,R0 & BNZ ILD159D  ;NO, JUMP; READ TOGGLE REG.
0142A 65050 F0085  9440             TSTRMSK R0,85                   ;DRIVE TYPE 5 (SWALLOW-4)?
0142B FFFFF 33C2D  9441             BNZ ILD159D                     ;NO, ASSUME SWALLOW-5
0142C 37351 F000A  9445             RDMSKB R1,H#A                   ;FLAG BYTE = TYPE 5 (SWALLOW-4)
                   9453   ILD159D:  JSR.P FLAGLOC                   ;LOC. OF DRIVE FLAGS
0142D E0401 1F342  9453 +           SETPGE FLAGLOC_SHR_H#B & JSRP FLAGLOC
0142E 3411D FFFFF  9455             ALDREG R1,RAM(                  ;SAVE DRIVE TYPE FLAG THIS DRIVE
0142F 0F10F F03E5  9456           ADDC DZ,FOUTL,ZERO,RAME & NOP LASTFLG;LAST DRIVE FLAG = 1
                   9457   ;
01430 3710F F024E  9458   ILD160:   RESETRAM CACHECNT               ;RESET RETRY COUNT
01431 3710F F0085  9459             RESETRAM CURCYL                 ;RESET CURRENT CYLINDER
01432 3535F F0008  9460             SETRBIT R15,8                   ;SET RTZ FLAG
01433 0F322 E115A  9461             CALL R2,ILDSK0                  ;RE-ZERO THE DRIVE
01434 0F322 E1239  9462             CALL R2,ILDRUN                  ;RE-SELECT THE DRIVE
01435 373F0 F0245  9463             RDRAMB R0,MAXCYLD(              ;GET MAX. DRIVE CYLINDER
01436 45058 F000E  9464             TSTRBIT R8,H#E                  ;ANY DRIVE TYPE FLAG BITS?
01437 FFFFF 33B4C  9465             BNZ ILD112B                     ;IF YES, RETURN
01438 FFFFF E13DE  9466             JMP ILD150A                     ;ELSE, TRY NEXT DRIVE TYPE
                   9467
                   9468
                   9469   TITLE ILD 2 - VERIFY THE PACK
```

Addr Line ILD 2 - VERIFY THE PACK

```
                    9471
                    9472
01439 37352 F1455   9473    ILD200:   RDMSKB R2,ILD210
0143A FFFFF 1F183   9483              JSR SAVCONT                    ;SAVE ILD CONTINUATION LOC.
0143B 0F322 E1239   9484              CALL R2,ILDRUN                 ;SELECT ILD DRIVE IF NEEDED
0143C C3020 FFFFF   9486              RD      SCP,MODESW,R0          ;$Read the Mode Switches
0143D 45050 F0002   9491              TSTRBIT R0,2                   ;NO ECC RETRIES?
0143E FFFFF 33440   9492              BOZ ILD201                     ;NO, CONTINUE
0143F 3535F F0020   9493              SETRBIT R15,20                 ;SET NO ECC CORRECTION FLAG
                    9494    ;
01440 FFFFF 1F1A7   9495    ILD201:   JSR CHKMSG1                    ;CHECK MSG. SENT 1
01441 FFFFF 33C4C   9496              BNZ ILD203                     ;IF SENT, SKIP MESSAGE
                    9497    ;
                    9498    ;   Start Verify
                    9499    ;
01442 37350 F0016   9500    ILD201A:  RDMSKB R0,D#22
01443 0F322 E0E7B   9501              CALL R2,ISSCMDA1               ;MSG: "VERIFYING DR. "
01444 0F322 E126E   9502              CALL R2,PRNTDRV                ;PRINT DRIVE NO.
01445 37350 F001C   9503              RDMSKB R0,D#28
01446 0F322 E0E7B   9504              CALL R2,ISSCMDA1               ;MSG: ", CYL. "
01447 FFFFF 1F22F   9505              JSR RECORD1                    ;GET THE CURRENT HEAD AND...
01448 373FB F00B5   9506              RDRAMB R11,CURCYL              ;CURRENT CYLINDER
01449 0F322 E1263   9507              CALL R2,DSPLY                  ;DISPLAY & PRINT CYL. & HEAD
                    9508    ;
0144A FFFFF 1F1A4   9509    ILD202:   JSR SETMSG1                    ;SET MESSAGE SENT 1 FLAG
0144B 0F322 E1239   9510              CALL R2,ILDRUN                 ;RE-SELECT DRIVE IF NECESSARY
0144C FFFFF 1F22F   9511    ILD203:   JSR RECORD1                    ;GET CURRENT HEAD AND...
0144D 373FB F00B5   9512              RDRAMB R11,CURCYL              ;CURRENT CYLINDER
                    9513    ;
0144E FFFFF 1F280   9514              JSR ILDCYHD                    ;DISPLAY & SAVE CYL. AND HEAD
                    9515    ;
0144F 0F322 E11FB   9554              CALL R2,CONV100
01450 377F7 F02D9   9555              OR DZ,LSBZ,RAME,R7 & NOP WRDSTRK
01451 337F7 FFFFF   9556              OR ZB,LSBZ,,R7                 ;WORDS TO VERIFY ONE TRACK
                    9557    ;
                    9558    ILD205:   EQU $
01452 3535F F0200   9562              SETRBIT R15,200                ;SET NO DATA XFER. FLAG
01453 5535E F0200   9563              RSTRBIT R14,200                ;RESET ALT. TRACK FLAG
01454 433F6 E0407   9564              RESETR R6 & JMP READ00         ;RESET 1ST WORD & READ ONE TRACK
```

Addr Line ILD 2 - VERIFY THE PACK

```
                    9566    ;
                    9567    ;    ILD CONTINUATION LOCATION
                    9568    ;
01455 5535F F0020   9569    ILD210:   RSTRBIT R15,20                    ;RESET NO ECC CORRECTION FLAG
01456 4505F F0400   9570              TSTRBIT R15,400                   ;CHECK SM RETRY FLAG
01457 603FF 3345B   9571              XOR AQ,B,R15,R15 & BOZ ILD211     ;NO, CONTINUE; YES, RESET IT
01458 FFFFF 1F22F   9572              JSR RECORD1                       ;GET CUR. HEAD AND...
01459 373FB F00B5   9573              RDRAMB R11,CURCYL                 ;CURRENT CYLINDER
0145A FFFFF E1440   9574              JMP ILD201                        ;TRY READING ONE MORE TIME
                    9575    ;
0145B 3710F F0018   9576    ILD211:   RESETRAM SEN415                   ;RESET SM-RETRY COUNT
0145C C3023 FFFFF   9577              RD SCP,MODESW,R3                  ;READ MODE SWITCHES
0145D 45053 F0002   9578              TSTRBIT R3,2                      ;MODE 2 (NO ECC RETRIES)?
0145E FFFFF 33460   9579              BOZ $+2                           ;NO, JUMP
0145F 3535F F0020   9580              SETRBIT R15,20                    ;SET NO ECC CORRECTION FLAG
                    9581    ;
01460 4505E F0200   9582              TSTRBIT R14,200                   ;CHECK ALT. TRACK FLAG
01461 323F2 FFFFF   9583              QTOR R2
01462 603EE 33467   9584              XOR AQ,B,R14,R14 & BOZ ILD212     ;IF NOT, JUMP; IF YES, RESET IT
01463 373F0 F00B9   9585              RDRAMB R0,NDCYL                   ;GET ENDING CYLINDER
01464 373F1 F020A   9586              RDRAMB R1,MAXCYLI                 ;GET MAX. USEABLE CYLINDER
01465 29110 FFFFF   9587              SUBB AB,FOUTL,R1,R0               ;IF END CYL. IS GREATER THAN...
01466 FFFFF 3547E   9588              CJMP ALB,ILD214                   ;MAX. CYL., VERIFY IS COMPLETE
                    9589    ;
01467 FFFFF 1F2EB   9590    ILD212:   JSR LIMIT                         ;CHECK FOR VERIFY DONE
01468 FFFFF 33C7E   9591              BNZ ILD214                        ;IF YES, JUMP
01469 4505F F1000   9592              TSTRBIT R15,1000                  ;CHECK ADV. CYLINDER FLAG
0146A 603FF 3346F   9593              XOR AQ,B,R15,R15 & BOZ ILD212A    ;IF NOT, JUMP; IF YES, RESET IT
0146B 353F2 F00BA   9594              OR DB,B,RAME,R2 & NOP ILDFLG      ;MERGE ALT. TRACK & ILD FLAGS
0146C 45052 F0210   9595              TSTRBIT R2,210                    ;IF FORMAT-TO-VERIFY FLAG OR...
0146D FFFFF 33C79   9596              BNZ ILD213                        ;ALT. TRACK FLAG, ADV. CYLINDER
0146E FFFFF E1472   9597              JMP ILD212B
                    9598    ;
0146F 2D1F1 F00BC   9599    ILD212A:  SUBB DB,FOUTL,RAME,R1 & NOP NDHD  ;IF END HEAD IS GREATER...
01470 FFFFF 34475   9600              CJMP AGB,ILD212C                  ;THAN CUR. HEAD, JUMP
01471 0F3FB F00B5   9601              ADD DZC,B,RAME,R11 & NOP CURCYL   ;INCR. NEXT CYLINDER
01472 FFFFF 1F22D   9602    ILD212B:  JSR STRTOVR1                      ;NEXT HEAD = START HEAD
01473 FFFFF 1F22F   9603              JSR RECORD1
01474 FFFFF E1479   9604              JMP ILD213                        ;SEEK TO NEXT CYLINDER
                    9605    ;
01475 5535F F0800   9606    ILD212C:  RSTRBIT R15,800                   ;RESET ADVANCE HEAD FLAG
                    9607              JSR.P ADVHEAD                     ;SET NEW HEAD
01476 E0400 1F51C   9607 +            SETPGE ADVHEAD_SHR_H#B & JSRP ADVHEAD
01477 341CF F00B6   9608              ALDRAM R12,CURHD                  ;SAVE CURRENT HEAD
01478 FFFFF E1440   9609              JMP ILD201
                    9610    ;
01479 0F322 E1263   9611    ILD213:   CALL R2,DSPLY                     ;DISPLAY CUR. CYL. & HEAD
0147A 0F322 E115A   9612              CALL R2,ILDSK0                    ;SEEK TO NEXT CYL.
0147B FFFFF 1F22F   9613              JSR RECORD1                       ;RETRIEVE CUR. HEAD AND...
0147C 373FB F00B5   9614              RDRAMB R11,CURCYL                 ;CURRENT CYLINDER
0147D 433F3 E1439   9615              RESETR R3 & JMP ILD200            ;VERIFY NEXT TRACK
                    9616    ;
                    9617    ILD214:   TSTMBIT 10,ILDFLG                 ;Format-to-Verify flag set?
```

```
 Addr            Line ILD 2 - VERIFY THE PACK

0147E 373F0 F00BA  9617 +          RDRAMB   R0,ILDFLG
0147F 45150 F0010  9617 +          TSTBIT   R0,10
01480 FFFFF 33C88  9618            BNZ ILD215                    ;IF SET, VERIFY IS COMPLETE
01481 45053 F0001  9619            TSTRBIT R3,)                  ;MODE 1 (CONTINUOUS READ)?
01482 FFFFF 33488  9620            BOZ ILD215                    ;IF NOT, VERIFY IS COMPLETE
                   9621    ;
                   9622            RSTMBIT MSG2+MSG1,MSGFLG      ;Delete Message Sent flags
01483 373F0 F01EF  9622 +          RDRAMB   R0,MSGFLG
01484 55350 F0006  9622 +          RSTRBIT  R0,MSG2+MSG1
01485 3410F F01EF  9622 +          ALDRAM   R0,MSGFLG
01486 FFFFF 1F22B  9623            JSR STRTOVR                   ;GET START CYL. & HEAD
01487 34308 E1473  9625            RTOR     R0,R11 & JMP ILD212B+1   ;$Start over
                   9631    ;
01488 35350 F0020  9632    ILD215: SETRBIT R0,20                 ;SET LAST OPERATION FLAG
01489 3410F F00BA  9633            ALDRAM R0,ILDFLG
0148A FFFFF E0092  9634            JMP NCMPLT0                   ;NORMAL COMPLETE ENDING
                   9635
                   9636
                   9637    TITLE ILD 3 - READ VERIFY ONE TRACK
                   9638
                   9639
                   9640    ;   DISPLAY THE LAST CYLINDER AND HEAD.
                   9641    ;
0148B 373F2 F00B8  9642    ILD300: RDRAMB R2,STCYL               ;GET LAST STARTING CYLINDER
0148C 373F1 F00B8  9643            RDRAMB R1,STHD                ;GET LAST STARTING HEAD
                   9644    ;
0148D 0F322 E12CA  9666            CALL R2,CKFLAG                ;CHECK FOR FLAG BYTE
0148E 373F2 F00B8  9667            RDRAMB R2,STCYL               ;GET STARTING CYL. AND...
0148F 373F1 F00BB  9668            RDRAMB R1,STHD                ;START HEAD FOR CHECKING
01490 3535E F0020  9669            SETRBIT R14,20                ;SET SPECIAL FLAG
01491 2D1F2 F020A  9670            SUBB DB,FOUTL,RAME,R2 & NOP MAXCYLI ;IF CYLINDER NUMBER IS...
01492 433F0 35499  9671            RESETR R0 & CJMP ALB,ILD304   ;GREATER THAN MAX., GET NEW CYL.
01493 2D1F1 F020D  9672            SUBB DB,FOUTL,RAME,R1 & NOP MAXHDI ;IF HEAD NO. IS GREATER...
01494 FFFFF 35499  9673            CJMP ALB,ILD304               ;THAN MAX., GET NEW ENTRIES
01495 5535E F0020  9674            RSTRBIT R14,20                ;RESET SPECIAL FLAG
                   9675    ;
01496 0F322 E1170  9676    ILD303: CALL R2,STRTCYL               ;PRINT CYLINDER NO. AND...
01497 0F322 E118D  9677            CALL R2,STRTHD                ;HEAD NUMBER
01498 FFFFF E1090  9678            JMP ILD06                     ;ARE THESE NUMBERS OK?
                   9679    ;
                   9680    ;   GET NEW CYLINDER AND HEAD PARAMETERS
                   9681    ;
01499 4505E F0020  9682    ILD304: TSTRBIT R14,20                ;SPECIAL FLAG SET?
0149A 603EE 33C9C  9683            XOR A0,B,R14,R14 & BNZ $+2    ;IF SET, RESET IT & JUMP
0149B 0F322 E12CA  9684            CALL R2,CKFLAG                ;CHECK FOR FLAG BYTE
0149C 433F0 FFFFF  9685            RESETR R0
0149D 0F322 E11A9  9686            CALL R2,GTSTCYL               ;GET CYLINDER NO. FROM OPERATOR
0149E 433F0 FFFFF  9687            RESETR R0
```

```
 Addr            Line ILD 3 - READ VERIFY ONE TRACK

0149F 0F322 E11C7  9688            CALL R2,GTSTHED                 ;GET HEAD NUMBER
                   9689    ;
014A0 37352 F14A2  9690    ILD306: RDMSKB R2,ILD310
014A1 FFFFF 1F183  9691            JSR SAVCONT                     ;SAVE ILD CONTINUATION LOC.
                   9692    ;
                   9693    ;   VERIFY ONE TRACK CONTINUATION LOCATION.
                   9694    ;
014A2 FFFFF 1F22B  9695    ILD310: JSR STRTOVR                     ;GET STARTING CYL. AND HEAD
014A3 5535F F1C00  9696            RSTRBIT R15,1C00                ;RESET SOME FLAGS
014A4 4505E F0200  9697            TSTRBIT R14,200                 ;ALT. TRACK FLAG?
014A5 603EE 33CAD  9698            XOR AQ,B,R14,R14 & BNZ ILD311   ;YES RESET IT & END THE ILD
014A6 FFFFF 1F1A7  9699            JSR CHKMSG1                     ;CHECK MSG. SENT 1
014A7 FFFFF 33C4C  9700            BNZ ILD203                      ;IF SET, SKIP MESSAGE
                   9701    ;
014A8 37350 F0016  9702            RDMSKB R0,D'22
014A9 0F322 E0E7B  9703            CALL R2,1SSCMDA1                ;MSG: "VERIFYING DR. "
014AA 0F322 E126E  9704            CALL R2,PRNTDRV                 ;PRINT DRIVE NO.
014AB 0F322 E1289  9705            CALL R2,CYLHED                  ;PRINT CYLINDER AND HEAD NOS.
014AC FFFFF E144A  9706            JMP ILD202                      ;VERIFY THE TRACK
                   9707    ;
014AD 373F0 F00BA  9708    ILD311: RDRAMB R0,ILDFLG                ;GET ILD FLAGS
014AE 35350 F0400  9709            SETRBIT R0,400                  ;SET THE FORMAT DEFECTIVE FLAG...
014AF FFFFF E1488  9710            JMP ILD215                      ;AND END THE ILD
                   9711
                   9712
                   9713    TITLE ILD 5 - OSCILLATING SEEK TEST
```

Addr Line ILD 5 - OSCILLATING SEEK TEST

```
                    9715
                    9716    ILD500:   EQU $
014B0 373F2 F00B8   9720              RDRAMB R2,STCYL                  ;GET LAST STARTING CYLINDER
014B1 373F1 F00B9   9721              RDRAMB R1,NDCYL                  ;GET LAST END CYLINDER
014B2 3710F F00B6   9722              RESETRAM CURHD                   ;CURRENT HEAD = 0
                    9723    ;
014B3 0F322 E12CA   9724              CALL R2,CKFLAG                   ;CHECK FOR FLAG BYTE
014B4 373F3 F020D   9725              RDRAMB R3,MAXHDI                 ;END HEAD = MAX. HEAD
014B5 373F2 F00B8   9726              RDRAMB R2,STCYL                  ;GET START CYLINDER AND...
014B6 373F1 F00B9   9727              RDRAMB R1,NDCYL                  ;END CYLINDER FOR CHECKING
014B7 2D1F2 F0245   9728              SUBB DB,FOUTL,RAME,R2 & NOP MAXCYLDI ;IF START CYLINDER > THAN...
014B8 433F0 354C1   9729              RESETR R0 & CJMP ALB,ILD503      ;MAX. DRIVE CYLINDER, JUMP
014B9 2D1F1 F0245   9730              SUBB DB,FOUTL,RAME,R1 & NOP MAXCYLDI ;IF END CYLINDER > THAN...
014BA 61121 35515   9731              XOR AB,FOUTL,R2,R1 & CJMP ALB,ILD700+1 ;MAX. DRIVE CYLINDER, JUMP
                    9732    ;
014BB FFFFF 334C1   9733    ILD502:   BOZ ILD503                       ;IF START = END CYL., JUMP
014BC 3710F F00BB   9734              RESETRAM STHD
014BD 3413F F00BC   9735              ALDRAM R3,NDHD                   ;SAVE ENDING HEAD
                    9736    ;
                    9737    ;   Print last Start & End cylinders
                    9738    ;
014BE 0F322 E1170   9739              CALL R2,STRTCYL                  ;DISPLAY STARTING CYLINDER
014BF 0F322 E1177   9740              CALL R2,ENDCYL                   ;DISPLAY ENDING CYLINDER
014C0 FFFFF E1090   9741              JMP ILD06                        ;ARE THESE NUMBERS OK?
                    9742    ;
014C1 0F322 E11A9   9743    ILD503:   CALL R2,GTSTCYL                  ;GET STARTING CYL. FROM OPERATOR
014C2 373F3 F020D   9744              RDRAMB R3,MAXHDI                 ;END HEAD = MAX. HEAD
014C3 433F0 FFFFF   9745              RESETR R0
                    9746    ;
014C4 3413F F00BC   9747              ALDRAM R3,NDHD                   ;SAVE ENDING HEAD
014C5 0F322 E11BA   9748              CALL R2,GTNDCYL                  ;GET END CYL. FROM OPERATOR
014C6 373F1 F00B8   9749              RDRAMB R1,STCYL                  ;GET START CYLINDER
                    9750    ;
014C7 61110 FFFFF   9751              XOR AB,FOUTL,R1,R0
014C8 FFFFF 331F6   9752              BOZ BADCYL                       ;IF START=END CYL., GET NEW ENTRY
014C9 FFFFF E1092   9753              JMP ILD06A
                    9754    ;
014CA 3710F F00B2   9755    ILD506:   RESETRAM MAINILD2                ;RESET START CYL. DISPLAY
014CB 3710F F00B3   9756              RESETRAM MAINILD3                ;RESET END CYL. DISPLAY
014CC 37350 F003C   9757              RDMSKB R0,D#60                   ;MSG: "OSCILLATING SEEK "
014CD 0F322 E120C   9758              CALL R2,ILDSTRT                  ;CHECK IF THIS IS O.K.
                    9759    ;
                    9760    ;   START OSCILLATING SEEK TEST
                    9761    ;
014CE 37352 F14D0   9762              RDMSKB R2,$+2
014CF FFFFF 1F183   9763              JSR SAVCONT                      ;SAVE ILD CONTINUATION LOC.
                    9764    ;
014D0 FFFFF 1F1A7   9765              JSR CHKMSG1                      ;CHECK MSG. SENT 1
014D1 FFFFF 33CD4   9766              BNZ ILD510                       ;IF SET, SKIP MESSAGE
                    9767    ;
014D2 37350 F003C   9768              RDMSKB R0,D#60
014D3 0F322 E1544   9769              CALL R2,SEEKRUN                  ;MSG: "OSCILLATING SEEK RUNNING "
```

```
 Addr            Line ILD 5 - OSCILLATING SEEK TEST

                 9770   ;
                 9771   ;   ILD-5 continuation loc.
                 9772   ;
014D4 373F1 F00BA 9773   ILD510:   RDRAMB R1,ILDFLG                ;GET ILD FLAGS
014D5 4505F F0400 9774             TSTRBIT R15,400                 ;CHECK SM-RETRY FLAG
014D6 FFFFF 33CDE 9775             BNZ ILD511+1                    ;IF SET, SEEK TO SAME CYLINDER
                 9776   ;
014D7 45051 F0100 9777             TSTRBIT R1,100                  ;CHECK FOR ALT. CYL. FLAG
014D8 60311 33CE1 9778             XOR A0,B,R1,R1 & BNZ ILD512     ;IF SET, RESET IT & JUMP
014D9 35351 F0100 9779             SETRBIT R1,100                  ;SET ALT. CYL. FLAG
014DA 3710F F00B3 9780             RESETRAM MAINILD3               ;CLEAR END CYL. DISPLAY
                 9781             MOVSMB STCYL,MAINILD2           ;Display Start Cylinder
014DB 373F0 F00B8 9781 +           RDRAMB   R0,STCYL
014DC 3410F F00B2 9781 +           ALDRAM   R0,MAINILD2
                 9782   ;
014DD 3410F F00B5 9783   ILD511:   ALDRAM R0,CURCYL                ;SAVE CYLINDER FOR SEEK
014DE 3411F F00BA 9784             ALDRAM R1,ILDFLG                ;SAVE ILD FLAGS
014DF 37352 F154B 9785             RDMSKB R2,READHDR               ;RETURN ADDRESS AFTER SEEK
014E0 FFFFF E115A 9786             JMP ILDSK0
                 9787   ;
014E1 3710F F00B2 9788   ILD512:   RESETRAM MAINILD2               ;CLEAR START CYLINDER DISPLAY
                 9789             MOVSMB NDCYL,MAINILD3           ;Display Ending Cylinder
014E2 373F0 F00B9 9789 +           RDRAMB   R0,NDCYL
014E3 3410F F00B3 9789 +           ALDRAM   R0,MAINILD3
014E4 FFFFF E14DD 9790             JMP ILD511
                 9791
                 9792
                 9793   TITLE ILD 6 - RANDOM SEEK TEST
```

Addr Line ILD 6 - RANDOM SEEK TEST

```
                    9795
                    9796    ILD600:   EQU $
014E5 0F322 E12CA   9801              CALL R2,CKFLAG                ;CHECK FOR DRIVE FLAG
014E6 373F0 F0245   9802              RDRAMB R0,MAXCYLD1            ;GET MAX. CYLINDER
014E7 373F1 F020D   9803              RDRAMB R1,MAXHD1              ;END HEAD = MAX. HEAD
014E8 3410F F00B9   9804    ILD601:   ALDRAM R0,NDCYL               ;END CYL.= MAX. CYL.
014E9 3411F F00BC   9805              ALDRAM R1,NDHD                ;SAVE ENDING HEAD
014EA 373F1 F00B5   9806              RDRAMB R1,CURCYL              ;GET CURRENT CYLINDER
014EB FFFFF 33CED   9807              BNZ $+2                       ;IF NOT ZERO, CUR. CYL.= SEED
014EC 34301 FFFFF   9808              RTOR R0,R1                    ;ELSE, MAX. CYL.= SEED WORD
014ED 3411F F01FD   9809              ALDRAM R1,SEEDWRD             ;SAVE THE INITIAL SEED WORD AND...
014EE 3411F F00B2   9810              ALDRAM R1,MAINILD2            ;DISPLAY IT
014EF 3710F F00B3   9811              RESETRAM MAINILD3
                    9812    ;
014F0 37350 F003D   9813              RDMSKB R0,D#61                ;MSG: "RANDOM SEEK "
014F1 0F322 E120C   9814              CALL R2,ILDSTRT               ;CHECK IF THIS IS O.K.
                    9815    ;
                    9816    ;   START RANDOM SEEK TEST
                    9817    ;
014F2 37352 F14F4   9818              RDMSKB R2,$+2
014F3 FFFFF 1F183   9819              JSR SAVCONT                   ;SAVE ILD CONTINUATION LOC.
                    9820    ;
014F4 FFFFF 1F1A7   9821              JSR CHKMSG1                   ;CHECK MESSGE SENT 1
014F5 433FC 33CF8   9822              RESETR R12 & BNZ ILD606       ;IF SET, SKIP MESSAGE
                    9823    ;
014F6 37350 F003D   9824              RDMSKB R0,D#61
014F7 0F322 E1544   9825              CALL R2,SEEKRUN               ;MSG: "RANDOM SEEK RUNNING "
                    9826    ;
                    9827    ;   ILD-6 continuation loc.
                    9828    ;
014F8 3710F F00B6   9829    ILD606:   RESETRAM CURHD                ;HEAD NO. IS ALWAYS 0
014F9 373FB F00B5   9830              RDRAMB R11,CURCYL             ;GET CUR. CYLINDER
014FA 4505F F0400   9831              TSTRBIT R15,400               ;CHECK SM-RETRY FLAG
014FB FFFFF 33D11   9832              BNZ ILD608                    ;IF SET, SEEK TO CUR. CYLINDER
014FC 3710F F00B3   9833              RESETRAM MAINILD3             ;CLEAR ILD DISPLAY 12 LSB'S
014FD 373FD F00B4   9834              RDRAMB R13,ILDDRV             ;GET ILD DRIVE NO.
014FE 5535D F0800   9835              RSTRBIT R13,800               ;RESET VALID BIT
                    9836              JSR.P ERRLOG                  ;GET ERROR/USAGE LOC. THIS DRIVE
014FF E0400 1F350   9836 +            SETPGE ERRLOG_SHR_H#B & JSRP ERRLOG
01500 0614E F0006   9837              ADD DQ,FOUTL,MSK1,IREG & NOP D#6   ;ADD BIAS FOR SEEK COUNT
                    9838              JSR.P SMBRD+1                 ;GET CURRENT SEEK COUNT
01501 E0400 1F03C   9838 +            SETPGE SMBRD+1_SHR_H#B & JSRP SMBRD+1
01502 34503 4F002   9839              OR AZ,RSBZ,R0,R3 & PSHLD 2
01503 335F3 887FF   9840              RSHFTZ R3                     ;MOVE IT INTO POSITION
01504 053F3 F002F   9841              ADD DB,B,RAME,R3 & NOP ATTNHIO     ;INCR.+ LAST ATTN. HIO NO.
01505 45353 F0FFF   9842              MSKRBIT R3,H#FFF              ;MASK IT TO 12 BITS
                    9843    ;
01506 37351 F063D   9844    ILD607:   RDMSKB R1,D#1597              ;MULTIPLIER = 1597 (DON'T ASK WHY)
01507 373F2 F01FD   9845              RDRAMB R2,SEEDWRD             ;TIMES SEED WORD
01508 0B3F3 1F204   9846              INCRB R3 & JSR MULTPLYB       ;INCR. SEEK COUNT
```

```
 Addr            Line ILD 6 - RANDOM SEEK TEST

                 9847    ;
01509 61330 FFFFF 9848            XOR AB,B,R3,R0                  ;XOR QUOTIENT WITH SEEK COUNT =
0150A 3410F F01FD 9849            ALDRAM R0,SEEDWRD               ;NEW SEED WORD
0150B 45350 F07FF 9853            MSKRBIT R0,7FF                  ;TRUNCATE QUOTIENT TO 11 BITS
                 9855    ;
0150C 2D1F0 F00B9 9856            SUBB DB,FOUTL,RAME,R0 & NOP NDCYL ;IF QUOTIENT > MAX. CYLINDER...
0150D 3430B 35506 9857            RTOR R0,R11 & CJMP ALB,ILD607   ;RE-COMPUTE CYLINDER NUMBER
                 9858    ;
0150E 650FB F00B5 9859            XOR DB,Q,RAME,R11 & NOP CURCYL  ;IF NEW CYLINDER = CUR...
0150F FFFFF 33506 9860            BOZ ILD607                      ;CYL., RE-COMPUTE CYL. NUMBER
                 9861    ;
01510 341BF F00B5 9862            ALDRAM R11,CURCYL               ;SET NEW CURRENT CYLINDER AND...
01511 341BF F00B2 9863    ILD608: ALDRAM R11,MAINILD2             ;DISPLAY IT
01512 37352 F154B 9864            RDMSKB R2,READHDR               ;RETURN LOC. AFTER SEEK
01513 FFFFF E115A 9865            JMP ILDSK0                      ;SEEK THE DRIVE
                 9866
                 9867
                 9868    TITLE ILD 7 - SEQUENTIAL SEEK TEST
```

Addr Line ILD 7 - SEQUENTIAL SEEK TEST

```
                    9870    ;
                    9871    ;  SET UP CYLINDER LIMITS
                    9872    ;
                    9873    ILD700:   EQU $
01514 3710F F00B8   9877              RESETRAM STCYL                ;STARTING CYLINDER = 0
                    9878    ;
01515 0F322 E12CA   9879              CALL R2,CKFLAG                ;CHECK FOR DRIVE FLAG
01516 373F0 F0245   9880              RDRAMB R0,MAXCYLD[            ;END CYL.= MAX. CYL.
01517 373F3 F020D   9881              RDRAMB R3,MAXHD1              ;END HEAD = MAX. HEAD
01518 3410F F00B9   9882    ILD701:   ALDRAM R0,NDCYL               ;SAVE ENDING CYLINDER
01519 3710F F00B6   9883              RESETRAM CURHD                ;CUR. HEAD NO. IS ALWAYS 0
0151A FFFFF E14BC   9884              JMP ILD502+1                  ;PRINT CYLINDER NUMBERS
                    9885    ;
0151B 37350 F003B   9886    ILD702:   RDMSKB R0,D#59                ;MSG: "SEQUENTIAL SEEK "
0151C 0F322 E120C   9887              CALL R2,ILDSTRT               ;CHECK IF THIS IS O.K.
                    9888    ;
                    9889    ;   START SEQUENTIAL SEEK TEST
                    9890    ;
0151D 373F0 F00B8   9891              RDRAMB R0,STCYL               ;GET START CYLINDER
0151E 373F1 F00B9   9892              RDRAMB R1,NDCYL               ;GET ENDING CYLINDER
0151F 29110 FFFFF   9893              SUBB AB,FOUTL,R1,R0           ;IF THE ENDING CYLINDER IS >...
01520 FFFFF 34526   9894              CJMP AGB,[LD704               ;THAN STARTING CYL., CONTINUE
                    9895    ;
01521 370FF F00BA   9896              RDRAMQ ILDFLG
01522 3605F F0100   9897              SETQBIT 100                   ;SET REV. SEEK FLAG
01523 321FF F00BA   9898              QLDRAM ILDFLG
01524 3410F F00B9   9899              ALDRAM R0,NDCYL               ;SET START CYL.= END CYL.
01525 3411F F00B8   9900              ALDRAM R1,STCYL               ;SET END CYL.= START CYL.
                    9901    ;
01526 3410F F00B2   9902    ILD704:   ALDRAM R0,MAINILD2            ;DISPLAY STARTING AND...
01527 3411F F00B3   9903              ALDRAM R1,MAINILD3            ;ENDING CYLINDERS
01528 37352 F152A   9904              RDMSKB R2,$+2
01529 FFFFF 1F183   9905              JSR SAVCONT                   ;SAVE ILD CONTINUATION LOC.
                    9906    ;
0152A FFFFF 1F1A7   9907              JSR CHKMSG1                   ;CHECK MESSAGE SENT 1
0152B FFFFF 33D30   9908              BNZ ILD705                    ;IF SET, SKIP MESSAGE
                    9909    ;
0152C 37350 F003B   9910              RDMSKB R0,D#59
0152D 0F322 E1544   9911              CALL R2,SEEKRUN               ;MSG: "SEQUENTIAL SEEK RUNNING "
                    9912    ;
0152E 37352 F154B   9913              RDMSKB R2,READHDR             ;RETURN ADDR. AFTER SEEK
0152F FFFFF E115A   9914              JMP ILDSK0                    ;SEEK TO STARTING CYLINDER
                    9915    ;
                    9916    ;   ILD-7 continuation loc.
                    9917    ;
01530 373F0 F00B5   9918    ILD705:   RDRAMB R0,CURCYL              ;GET CURRENT CYLINDER
01531 373F1 F00BA   9919              RDRAMB R1,ILDFLG              ;GET ILD FLAGS
01532 4505F F0400   9920              TSTRBIT R15,400               ;CHECK SM-RETRY FLAG
01533 FFFFF 33CDD   9921              BNZ ILD511                    ;IF SET, SEEK TO CUR. CYLINDER
                    9922    ;
01534 45051 F0100   9923              TSTRBIT R1,100                ;CHECK REV. SEEK FLAG
01535 FFFFF 33D3C   9924              BNZ ILD707                    ;IF SET, JUMP
```

```
 Addr            Line ILD 7 - SEQUENTIAL SEEK TEST

                 9925   ;
01536 2D1F0 F00B9 9926           SUBB DB,FOUTL,RAME,R0 & NOP NDCYL  ;IF END CYLINDER IS GREATER...
01537 FFFFF 34541 9927           CJMP AGB,ILD708                    ;THAN CURRENT CYL., FORWARD SEEK
                 9928   ;
                 9929   ;   START REVERSE SEEKS
                 9930   ;
01538 370FF F00B8 9931           RDRAMQ STCYL                       ;DISPLAY STARTING CYLINDER...
01539 321FF F00B3 9932           QLDRAM MAINILD3                    ;AS FINAL SEEK DESTINATION
0153A 35351 F0100 9933           SETRBIT R1,100                     ;SET REV. SEEK FLAG
                 9934   ;
0153B 133F0 E1542 9935   ILD706: DECRB R0 & JMP ILD710              ;DECR. CURRENT CYLINDER
                 9936   ;
0153C 2D1F0 F00B8 9937   ILD707: SUBB DB,FOUTL,RAME,R0 & NOP STCYL   ;IF START CYLINDER IS LESS...
0153D FFFFF 3553B 9938           CJMP ALB,ILD706                    ;THAN CURRENT CYL., REVERSE SEEK
                 9939   ;
                 9940   ;   START FORWARD SEEKS
                 9941   ;
0153E 370FF F00B9 9942           RDRAMQ NDCYL                       ;DISPLAY ENDING CYLINDER...
0153F 321FF F00B3 9943           QLDRAM MAINILD3                    ;AS FINAL SEEK DESTINATION
01540 55351 F0100 9944           RSTRBIT R1,100                     ;RESET REV. SEEK FLAG
                 9945   ;
01541 0B3F0 FFFFF 9946   ILD708: INCRB R0                           ;INCR. CURRENT CYLINDER
                 9947   ;
01542 3410F F00B2 9948   ILD710: ALDRAM R0,MAINILD2                 ;DISPLAY CURRENT CYLINDER
01543 FFFFF E14DD 9949           JMP ILD511                         ;SEEK THE DRIVE
                 9950
                 9951
                 9952   ;   THIS ROUTINE PRINTS THE 'RUNNING' MESSAGE FOR SEEK ILD'S.
                 9953   ;Upon entry:  R0 = message no. to be printed.
                 9954
01544 FFFFF 1F2A3 9955   SEEKRUN: JSR SAVRET4                       ;SAVE RETURN FROM THIS ROUTINE
01545 0F322 E0E7B 9956           CALL R2,ISSCMDA1                   ;PRINT SEEK MESSAGE
01546 37350 F0043 9957           RDMSKB R0,D#67
01547 0F322 E0E7B 9958           CALL R2,ISSCMDA1                   ;PRINT "RUNNING DR. "
01548 0F322 E126E 9959           CALL R2,PRNTDRV                    ;PRINT THE DRIVE NUMBER
01549 FFFFF 1F1A4 9960           JSR SETMSG1                        ;SET MESSAGE SENT 1 FLAG...
0154A FFFFF E12A0 9961           JMP RTRN4                          ;AND RETURN
                 9962
                 9963
                 9964   TITLE SEEK TESTS  -  HEADER VERIFY
```

Addr Line SEEK TESTS - HEADER VERIFY

```
                    9966
                    9967    ;    This is the routine used in ILD Seek tests to verify drive position when
                    9968    ;a seek completes.  The routine is enabled with Mode Switch 1.  All seeks are
                    9969    ;verified on Head 0.  Seeks to Alternate cylinders are not verified.
                    9970    ;The routine is entered from ILD24 upon completion of a Seek.
                    9971    ;Return is to the location saved in SAVCONT after going thru the Wait Loop
                    9972    ;via ILD20.
                    9973    ;Upon entry:  CURCYL = Desired cylinder
                    9974    ;
0154B C3020 FFFFF   9975    READHDR:  RD SCP,MODESW,R0                  ;READ THE MODE SWITCHES
0154C 45350 F000F   9976              MSKRBIT R0,H#F                    ;ISOLATE THE SWITCHES
0154D 45050 F0001   9977              TSTRBIT R0,1                      ;CHECK FOR MODE 1
0154E FFFFF 33570   9978              BOZ READHDR5                      ;IF NOT, RETURN
                    9979    ;
0154F 373FB F00B5   9980              RDRAMB R11,CURCYL                 ;GET THE CURRENT CYLINDER
01550 370FF F020A   9981              RDRAMQ MAXCYLI                    ;GET MAX. CYLINDER
01551 281B0 FFFFF   9982              SUBB AQ,FOUTL,R11,ZERO            ;IF THE CURRENT CYLINDER IS...
01552 FFFFF 34570   9983              CJMP AGB,READHDR5                 ;GREATER THAN MAX., RETURN
                    9984    ;
01553 0F322 E1239   9985              CALL R2,ILDRUN                    ;RE-SELECT THE ILD DRIVE
01554 433F5 FFFFF   9986              RESETR R5                         ;RESET READ/WRITE FLAGS
01555 373FB F00B5   9987    READHDR1: RDRAMB R11,CURCYL                 ;READ CURRENT CYLINDER
01556 34383 FFFFF   9988              RTOR R11,R3                       ;MOVE CYLINDER TO R3
01557 433F9 FFFFF   9989              RESETR R9                         ;RECORD NO. IS 0
01558 433FC FFFFF   9990              RESETR R12                        ;HEAD NO. IS 0
                    9991              JSR.P DRVHDR                      ;LOAD THE EXPECTED HEADER
01559 E0403 1F569   9991 +            SETPGE DRVHDR_SHR_H#B & JSRP DRVHDR
0155A B1F00 F0030   9992              WRTE HSSM,LDXMCJ & NOP RDHDR
0155B E100D FFFFF   9993              CTL HSSM,XMCJCMD                  ;START READ HEADER
                    9994    ;
0155C 4505F F0400   9995              TSTRBIT R15,400                   ;SM-RETRY FLAG?
0155D FFFFF 33D60   9996              BNZ $+3                           ;IF SET, ALLOW NORMAL RECOVERY
                    9997              SETMBIT D#10,SEN415               ;Retry = 10 to prevent recovery
0155E 37350 F000A   9997 +            RDMSKB   R0,D#10
0155F 3410F F0018   9997 +            ALDRAN   R0,SEN415
01560 35355 F0100   9998              SETRBIT R5,100                    ;SET SM-READ ACTIVE
01561 37350 F1572   9999              RDMSKB R0,READHDR6                ;SM-ERROR RETURN LOC.
                   10000              JSR.P SAVSMRTN                    ;SAVE IT IS CASE OF ERROR
01562 E0400 1F4C6  10000 +            SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
01563 37350 F61A8  10001              RDMSKB R0,61A8                    ;SET 50-MS. TIMER
                   10002              JSR.P SAVSMTIM                    ;SAVE IT
01564 E0400 1F4C2  10002 +            SETPGE SAVSMTIM_SHR_H#B & JSRP SAVSMTIM
01565 37350 F156B  10003              RDMSKB R0,READHDR4                ;SM-DONE RETURN LOC.
                   10004              JSR.P SAVIRPT                     ;SAVE IT
01566 E0400 1F4C0  10004 +            SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
01567 3535F F0001  10005              SETRBIT R15,1                     ;SET HEADER READ FLAG
                   10006    ;
                   10007    READHDR3: JSR.P SMIRPT                      ;CHECK FOR AN INTERRUPT
01568 E0400 1F37C  10007 +            SETPGE SMIRPT_SHR_H#B & JSRP SMIRPT
```

Addr Line SEEK TESTS - HEADER VERIFY

```
                10008   ;
                10009           JSR.P SMTIMER                   ;DECR. TIMER AND...
01569 E0400 1F4DB 10009 +       SETPGE SMTIMER_SHR_H#B & JSRP SMTIMER
0156A FFFFF E1568 10010         JMP READHDR3                    ;CHECK AGAIN
                10011   ;
                10012   ;   NORMAL COMPLETE
                10013   ;
0156B E1000 FFFFF 10014 READHDR4: CTL HSSM,DIFRST
                10015           JSR.P HSWAITR                   ;WAIT LOOP ROUTINE
0156C E0400 1F525 10015 +       SETPGE HSWAITR_SHR_H#B & JSRP HSWAITR
0156D 3710F F0018 10016         RESETRAM SEN415                 ;RESET SM-RETRY COUNT
0156E FFFFF 1F579 10017         JSR READCYL                     ;READ THE ACTUAL CYLINDER
0156F FFFFF 33D76 10018         BNZ READHDR7                    ;IF #COMPARE, ERROR
01570 5535F F0401 10019 READHDR5: RSTRBIT R15,401               ;RESET RETRY/HEADER READ FLAGS
01571 FFFFF E10D3 10020         JMP ILD20+1                     ;NORMAL EXIT
                10021   ;
                10022   ;   SM-ERROR RETURN
                10023   ;
01572 4505E F0200 10024 READHDR6: TSTRBIT R14,200               ;CHECK FOR ALT. TRACK FLAG
01573 603EE 33D55 10025         XOR AQ,B,R14,R14 & BNZ READHDR1 ;IF SET, RESET IT & VERIFY ALT.
                10026   ;
01574 FFFFF 1F579 10027         JSR READCYL                     ;READ THE ACTUAL CYLINDER
01575 FFFFF 330D2 10028         BOZ ILD20                       ;IF CYL. OK, WAIT LOOP & RETRY
01576 3410F F00B3 10029 READHDR7: ALDRAM R0,MAINILD3            ;DISPLAY ACTUAL CYLINDER
01577 3535F F8000 10030         SETRBIT R15,8000                ;SET SEEK ERROR FLAG
01578 FFFFF E10F8 10031         JMP ILD25B                      ;ERROR RETURN
                10032
                10033   ;Registers used: R0,R1.
                10034   ;Upon exit:  R0 = Actual cylinder
                10035   ;            R1-upper = Cylinder 4 msb's
                10036   ;            ALU = 0 if actual cylinder = expected cylinder (R11).
                10037
01579 C10A1 FFFFF 10038 READCYL:  RD HSSM,RDDRCYLM,R1
0157A C10A1 C0003 10039         RD HSSM,RDDRCYLM,R1 & LDCT 3    ;READ CYL. MSB'S TO R1
0157B 45351 F000F 10040         MSKRBIT R1,H#F                  ;DELETE THE DRIVE NUMBER AND...
0157C 01711 9857C 10041         ADD AB,LSBZ,R1,R1 & CJPCNT EVCNTZ,$ ;MOVE CYL. MSB'S INTO POSITION
0157D C1090 FFFFF 10042         RD HSSM,RDDRCYLL,R0
0157E C1090 FFFFF 10043         RD HSSM,RDDRCYLL,R0             ;READ CYL. LSB'S TO R0
0157F 31310 4F001 10044         OR AB,B,R1,R0 & PSHLD 1         ;COMPLETE CYLINDER TO R0
                10045           LSHFT2 R1,R1                    ;CYL. MSB'S TO R1-UPPER
01580 01711 887FF 10045 +       ADD AB,LSBZ,R1,R1 & CLPCNT EVCNTZ
01581 611B0 AF7FF 10046         XOR AB,FOUTL,R11,R0 & RTS       ;ACTUAL CYL.= EXPECTED?  RETURN
                10047
                10048
                10049   TITLE ILD 8 & 9 - ECC DISPLAY CLEAR
```

Addr Line ILD 8 & 9 - ECC DISPLAY CLEAR

```
                10051
                10052   ;          ILD 8 - DISK ECC DISPLAY CLEAR
                10053   ;
01582 37350 F003F 10054   ILD800:   RDMSKB R0,D#63                  ;MSG: "DISK ECC DISPALY CLEAR."
01583 0F322 E1214 10055             CALL R2,ILDSTRTA                ;IS THIS O.K?
                10056   ;
01584 3714E F0093 10057             MOVE MSK1,IREG & NOP H#93       ;1ST LOC. TO CLEAR
01585 FFFFF C0014 10058             LOCT 20
                10059               JSR.P CLR25                     ;CLEAR ECC HISTORY DISPLAYS
01586 E0400 1F33D 10059 +           SETPGE CLR25_SHR_H#B & JSRP CLR25
                10060   ;
01587 3705F F000F 10061             RDMSKQ H#F
01588 0F322 E0E7C 10062             CALL R2,ISSCMDA                 ;CLEAR MESSAGE DISPLAY
01589 37350 F003F 10063             RDMSKB R0,D#63                  ;MSG: "DISK ECC DISPLAY CLEAR."
0158A 0F322 E0E7B 10064   ILD802:   CALL R2,ISSCMDA1                ;PRINT THE MESSAGE
0158B 37350 F002A 10065             RDMSKB R0,D#42
0158C 0F322 E0E7B 10066             CALL R2,ISSCMDA1                ;MSG: " COMPLETE"
0158D FFFFF E1135 10067             JMP ILD29                       ;EXIT THE ILD
                10068
                10070   TITLE ILD-B - CDC SPECIAL
```

Addr Line ILD-B - CDC SPECIAL

```
                10072
                10073   ;    Equates for this ILD
                10074   ;
                10075   MES1:     EQU D#66                    ;$Message: "TEST."
                10076   MES2:     EQU D#52                    ;$" CODE=" message
                10077   ERSCMD:   EQU H#F                     ;$Clear OCP display command
                10078   RUNMES:   EQU D#67                    ;$"RUNNING DR. " message
                10079   GETHEX:   EQU H#86                    ;$Get Hex digits OCP command
                10080   PRTHEX:   EQU H#8                     ;$Print Hex digits command
                10081   ;
                10082   ;_____________________________________________________________________
                10083   ;
0158E 37350 F0006 10084   CDCILD:   RDMSKB   R0,6             ;$Cursor position = 6
0158F 3705F F000A 10085             RDMSKQ   H#A              ;$Position Cursor command
01590 0F322 E0E7C 10086             CALL     R2,ISSCMDA       ;$Do it
                10087   ;
01591 37350 F0042 10088             RDMSKB   R0,MES1          ;$Message: "TEST."
01592 0F322 E0E7B 10089             CALL     R2,ISSCMDA1      ;$Print it
                10090   ;
01593 37350 F0034 10091             RDMSKB   R0,MES2          ;$Message: " CODE="
01594 0F322 E0E7B 10092             CALL     R2,ISSCMDA1      ;$Print it
                10093   ;
01595 37350 F0045 10094             RDMSKB   R0,D#69          ;$Message: "?"
01596 0F322 E0E7B 10095             CALL     R2,ISSCMDA1      ;$Print it
                10096   ;
                10097   ;   Get 3 hex digits from the operator.
                10098   ;
01597 3705F F000A 10099             RDMSKQ   H#A              ;$Position the cursor...
01598 37350 F0011 10100             RDMSKB   R0,D#17          ;$to column 17
01599 0F322 E0E7C 10101             CALL     R2,ISSCMDA       ;$Do it
                10102   ;
0159A 3705F F0086 10103             RDMSKQ   GETHEX           ;$Command to get Hex digits
0159B 37350 F0003 10104             RDMSKB   R0,3             ;$Three digits needed
0159C 0F322 E0E7C 10105             CALL     R2,ISSCMDA       ;$Get them from the operator
0159D 3410F F0236 10106             ALDRAM   R0,SCRATCH       ;$Temporarily save them
                10107   ;
0159E 373F0 F0303 10108             RDRAMB   R0,FRBUNIT       ;$Any previous ILD number?
0159F FFFFF 33DB1 10109             BNZ      CDCILD2          ;$Yes, skip messages
015A0 3705F F000F 10110             RDMSKQ   ERSCMD           ;$Clear OCP display
015A1 0F322 E0E7C 10111             CALL     R2,ISSCMDA       ;$Do it
015A2 37350 F0037 10112             RDMSKB   R0,D#55          ;$Message: "ILD "
015A3 0F322 E0E7B 10113             CALL     R2,ISSCMDA1      ;$Print it
                10114   ;
015A4 373F0 F0236 10115             RDRAMB   R0,SCRATCH       ;$Get the saved digits
015A5 37351 F0003 10116             RDMSKB   R1,3             ;$There are 3 of them
015A6 3705F F0008 10117             RDMSKQ   PRTHEX           ;$Command to print hex chars.
015A7 0F322 E0E7C 10118             CALL     R2,ISSCMDA       ;$Do it
                10119   ;
015A8 3705F F000A 10120             RDMSKQ   H#A              ;$Position the cursor...
015A9 37350 F0008 10121             RDMSKB   R0,8             ;$to Position 8
015AA 0F322 E0E7C 10122             CALL     R2,ISSCMDA       ;$Do it
015AB 37350 F0043 10123             RDMSKB   R0,RUNMES        ;$Message: "RUNNING DR. "
015AC 0F322 E0E7B 10124             CALL     R2,ISSCMDA1      ;$Print it
                10125   ;
```

Addr Line ILD-B - CDC SPECIAL

```
                10126            MOVSMB   SCRATCH,FRBUNIT          ;$Move the ILD code to FRBUNIT
015AD 373F0 F0236 10126 +          RDRAMB   R0,SCRATCH
015AE 3410F F0303 10126 +          ALDRAM   R0,FRBUNIT
015AF 45350 F001F 10127            MSKRBIT  R0,1F                  ;$Isolate the Drive no. and...
015B0 0F322 E1270 10128            CALL     R2,PRTNBR              ;$Print it
                10129    ;
                10130    ;   Send code to the HSP interface.
                10131    ;
015B1 E407E FFFFF 10132    CDCILD2:  CTL      HIO,SPILCHI            ;$Select the HSP interface
015B2 E46F1 FFFFF 10133            CTL      HIO,ENCHT              ;$Assure tags are enabled
015B3 771FA F0236 10134            XNOR DZ,FOUTL,RAME,UBR1 & NOP SCRATCH ;$Load 3 hex digits to lsb's
015B4 7715B F03C0 10135            XNOR DZ,FOUTL,MSK2,UBR2 & NOP H#3C0 ;$Load Recovery Action code = F
015B5 7710C FFFFF 10136            XNOR     DZ,FOUTL,ZERO,UBR3     ;$ 12 msb's are 0's
015B6 E4401 FFFFF 10137            CTL      HIO,MDRVGTS            ;$Enable word onto the U-bus and...
015B7 E417B FFFFF 10138            CTL      HIO,SETEI              ;$set E.I.
                10139    ;
                10140    ;   Wait for response.
                10141    ;
015B8 37350 F0727 10142            RDMSKB   R0,727                 ;$Set 3-second...
015B9 D0FF0 FFFFF 10143            RDXFR    SMC,,R0                ;$timer into...
015BA 35350 F00E0 10144            SETRBIT  R0,H#E0                ;$ R0
015BB 37351 F1004 10145            RDMSKB   R1,1004                ;$Mask to allow IA or EF
                10146    ;
015BC 45081 335BE 10147            AND      DB,0,HIOST,R) & BOZ $+2  ;$If time-out, jump
015BD 133F0 335BC 10148            DECRB    R0 & BOZ $-1           ;$Decr. timer;  No irpt., jump back
                10149    ;
                10150    ;   Interrupt received or 3-second timeout.
                10151    ;
015BE E0204 FFFFF 10152            CTL      SMC,RSTIDAK            ;$Acknowledge the interrupt
015BF E0090 4F7FF 10153            CTL      SMC,UMPRST & PSHLD H#7FF ;$De-select the HSP
015C0 FFFFF FFFFF 10154            NOP                             ;$Delay here while...
015C1 FFFFF 887FF 10155            CLPCNT   EVCNTZ                 ;$the E.I. is accepted
015C2 FFFFF E113B 10156            JMP      )LD29A+)               ;$Return to ILD code
                10157
                10158    TITLE DISPLAY CDC TEST RESULTS
```

Addr Line DISPLAY CDC TEST RESULTS

```
                10160
                10161   ;    Upon entry:  R7 = code no. from NO-OP function.
                10162   ;
015C3 3705F F000F 10163   DISCDE:   RDMSKQ   ERSCMD                  ;$Clear OCP display
                10164             JSR.P    ISSCMD                  ;$Do it
015C4 E0401 1F624 10164 +         SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
                10165   ;
                10166   ;  Check for "ILD Terminated"
                10167   ;
015C5 65057 F0FF9 10168           TSTRMSK  R7,H#FF9                  ;$"ILD Terminated" code?
015C6 FFFFF 33DD3 10169           BNZ      DISCDE2                   ;$If not, jump
015C7 37350 F001B 10170           RDMSKB   R0,D#27                   ;$Msg: "ILD HALTED @ CYL. "
                10171   ;
                10172   DISCDE1:  JSR.P    ISSMESS                   ;$Print the message
015C8 E0403 1F0D0 10172 +         SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
015C9 373F0 F0303 10173           RDRAMB   R0,FRBUNIT                ;$Get the ILD code
015CA 45050 F001F 10174           TSTRBIT  R0,1F                     ;$Isolate the Drive no. in Q
015CB 0615E F0030 10175           ADD DQ,FOUTL,MSK2,IREG & NOP H#30  ;$Loc. of cylinder this drive
015CC 373D0 FFFFF 10176           RDREGB   RAMI,R0                   ;$Get current cylinder
015CD FFFFF 1F279 10177           JSR      CHARCNT                   ;$Compute character count
015CE 3705F F0009 10178           RDMSKQ   9                         ;$Command to print Decimal no.
                10179             JSR.P    ISSCMD                    ;$Print the current cylinder
015CF E0401 1F624 10179 +         SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
                10180   ;
                10181   ;  Exit from this code
                10182   ;
015D0 3710F F0303 10183   DISCDE1A: RESETRAM FRBUNIT                 ;$Reset ILD code
                10184             JSR.P    DOUBLE                    ;$Double drive no. if needed
015D1 E0403 1F125 10184 +         SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
015D2 FFFFF E0092 10185           JMP      NCMPLT0                   ;$Normal ending status
                10186   ;
                10187   ;  Check for "ILD Complete"
                10188   ;
015D3 65057 F0FFA 10189   DISCDE2:  TSTRMSK  R7,H#FFA                 ;$"ILD Complete" code?
015D4 333F7 33DDC 10190           OR       ZB,B,,R7 & BNZ DISCDE4    ;$If not, check further
015D5 37350 F0037 10191           RDMSKB   R0,D#55                   ;$Message: "ILD "
                10192             JSR.P    ISSMESS                   ;$Print it
015D6 E0403 1F0D0 10192 +         SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
                10193   ;
015D7 FFFFF 1F5F0 10194           JSR      DISCODE                   ;$Print the ILD code no.
                10195   ;
015D8 37350 F002A 10196           RDMSKB   R0,D#42                   ;$Message: " COMPLETE"
                10197             JSR.P    ISSMESS                   ;$Print it
015D9 E0403 1F0D0 10197 +         SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
                10198   ;
015DA 37350 F001C 10199   DISCDE3:  RDMSKB   R0,D#28                 ;$Message: ", CYL. "
015DB FFFFF E15C8 10200           JMP      DISCDE1                   ;$Print it & cylinder no.
                10201   ;
```

```
 Addr            Line DISPLAY CDC TEST RESULTS

015DC FFFFF 335E8 10202   DISCDE4:  BOZ      DISCDE5                ;$If no code, jump
015DD 45057 F0FE0 10203             TSTRBIT  R7,H#FE0               ;$Isolate code 7 msb's
015DE 6605F F0020 10204             TSTQMSK  20                     ;$Message code 2x or 3x?
015DF FFFFF 335EA 10205             BOZ      DISCDE6                ;$Yes, jump
                  10206   ;
                  10207   ;  Print ILD-B Error Message.
                  10208   ;
015E0 37350 F0037 10209             RDMSKB   R0,D#55                ;$Message: "ILD "
                  10210             JSR.P    ISSMESS                ;$Print it
015E1 E0403 1F0D0 10210 +           SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
                  10211   ;
015E2 FFFFF 1F5F0 10212             JSR      DISCODE                ;$Print the ILD code
                  10213   ;
015E3 37350 F0034 10214             RDMSKB   R0,MES2                ;$Message: " CODE="
                  10215             JSR.P    ISSMESS                ;$Print it
015E4 E0403 1F0D0 10215 +           SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
015E5 34370 FFFFF 10216             RTOR     R7,R0                  ;$Error code to R0
015E6 FFFFF 1F5F1 10217             JSR      DISCODE+1              ;$Print it
015E7 FFFFF E15DA 10218             JMP      DISCDE3                ;$Print the cylinder no.
                  10219   ;
015E8 37350 F0036 10220   DISCDE5:  RDMSKB   R0,D#54                ;$Message: "CHANNEL READY"
015E9 FFFFF E15EB 10221             JMP      $+2
015EA 37350 F0042 10222   DISCDE6:  RDMSKB   R0,MES1                ;$Message: "TEST."
                  10223             JSR.P    ISSMESS                ;$Print the message
015EB E0403 1F0D0 10223 +           SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
                  10224   ;
015EC 37350 F0034 10225             RDMSKB   R0,MES2                ;$Message: " CODE="
                  10226             JSR.P    ISSMESS                ;$Print it
015ED E0403 1F0D0 10226 +           SETPGE ISSMESS_SHR_H#B & JSRP ISSMESS
015EE 34370 1F5F1 10227             RTOR     R7,R0 & JSR DISCODE+1  ;$Print the code number
015EF FFFFF E15D0 10228             JMP      DISCDE1A               ;$Send Ending status
                  10229   ;
015F0 373F0 F0303 10230   DISCODE:  RDRAMB   R0,FRBUNIT             ;$Get the ILD code
015F1 37351 F0003 10231             RDMSKB   R1,3                   ;$Print three...
015F2 3705F F0008 10232             RDMSKQ   PRTHEX                 ;$hex characters
015F3 FFFFF E0E24 10233             JMP      ISSCMD                 ;$Print the code & return
                  10541   ;  ;Select Condensed Graphics print mode
                  10542   TITLE LINK TABLES
```

Addr Line LINK TABLES

```
                10544   ;╔═══════════════════════════════════════════════════════════╗
                10545   ;|                LINK TABLE FOR MAIN                        |
                10546   ;╚═══════════════════════════════════════════════════════════╝
                10566   ;
                10567   ;       From MAIN to DIAG4A
                10568   ;=========================================================
                10569   DIAGI:    EQU DIAG + 0
                10604   ;
01800           10605   ORG MAIN                            ;256 Locations reserved for linkages
                10606
                10607   ;       From DIAG4A into MAIN
01810           10608   ORG DIAGMAIN                                ;Linkage: Diagnostics-to-Main     (16 locs.)
                10609   ;=========================================================
01810 FFFFF E0E24 10610           JMP     ISSCMD                    ; + 0
01811 FFFFF FFFFF 10612           NOP                               ;$ +1 not used
01812 FFFFF E18D0 10616           JMP     ISSMESS                   ; + 2
01813 FFFFF E1880 10617           JMP     PRINT3                    ; + 3
01814 FFFFF E0E7C 10618           JMP     ISSCMDA                   ; + 4
01815 FFFFF E1B77 10619           JMP     DIAGRTN                   ; + 5
                10700
                10701   ;=====================================================================
01880           10702   ORG H$1880
                10703   ;
                10704   ;    This routine will print 3 hex characters on the OCP display.  All OCP
                10705   ;errors will be ignored so the routine will always return.
                10706   ;Upon entry:  R0 = 3 hex digits to be printed.
                10707   ;             R2 = Return address.
                10708   ;Registers used: R0,R1,Q & IREG.
                10709   ;
01880 3492F F01F8 10710   PRINT3:   ALDRAMU R2,RETURN1A               ;Save return msb's and...
01881 3412F F01EC 10711           ALDRAM R2,RETURN1                 ;return lsb's
01882 37351 F0003 10712           RDMSKB R1,3                       ;Print three...
01883 3705F F0008 10713           RDMSKQ 8                          ;hex characters
                10714             JSR.P ISSCMD                      ;Do it...
01884 E0401 1F624 10714 +         SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
01885 FFFFF E0E8E 10715           JMP RTRN1                         ;and return
                10716   ;
                10717   TITLE E.F. COMMAND DECODER
```

```
 Addr                Line E.F. COMMAND DECODER


                     10719
01890                10720     ORG H#1890
                     10721     ;
                10722  ;┌──────────────────────────────────────────────────────────────
                10723  ;| PURPOSE: This is the the EF command decoder jump table which
                10724  ;|       replaces the mapping PROM on the XMC.
                10725  ;└──────────────────────────────────────────────────────────────
                     10726                                        ;table EFCMDEC
                     10727     EFCMDEC: EQU $                     ;CASE OF sub_cmd
01890 FFFFF E0100    10729               JMP     COMREJ1          ;|$00 Read I.P.L. is invalid
01891 FFFFF E00A7    10733               JMP     PREPTRAK         ;| 01 PREP TRACK
01892 FFFFF E0100    10734               JMP     COMREJ1          ;| 02
01893 FFFFF E15C3    10736               JMP     DISCDE           ;|$03 Display Diagnostic test result
01894 FFFFF E0096    10740               JMP     SENSIO           ;| 04 SENSE I/O
01895 FFFFF E0100    10744               JMP     COMREJ1          ;| 05 'Test Write': Invalid
01896 FFFFF E0100    10746               JMP     COMREJ1          ;| 06
01897 FFFFF E009D    10747               JMP     SEEK             ;| 07 SEEK
01898 FFFFF E0100    10748               JMP     COMREJ1          ;| 08
01899 FFFFF E00A1    10750               JMP     WRDATA           ;|$09 Same as Write Data
0189A FFFFF E00B6    10754               JMP     RDVERIFY         ;| 0A READ VERIFY
0189B FFFFF E009A    10755               JMP     RELSEEK          ;| 0B TRACK POSITION RELATIVE SEEK
0189C FFFFF E0100    10756               JMP     COMREJ1          ;| 0C
0189D FFFFF E00A1    10757               JMP     WRDATA           ;| 0D WRITE DATA
0189E FFFFF E00B7    10758               JMP     READDATA         ;| 0E READ DATA
0189F FFFFF E0100    10759               JMP     COMREJ1          ;| 0F
018A0 FFFFF E0100    10760               JMP     COMREJ1          ;| 10
018A1 FFFFF E0100    10761               JMP     COMREJ1          ;| 11
018A2 FFFFF E00C3    10762               JMP     READCNT          ;| 12 READ COUNT
018A3 FFFFF E0098    10763               JMP     CALIBRAT         ;| 13 RECALIBRATE
018A4 FFFFF E0100    10765               JMP     COMREJ1          ;|$14 Read Buffered Record is invalid
018A5 FFFFF E0100    10766               JMP     COMREJ1          ;|$15 Write Home Address and R0 is Invali
018A6 FFFFF E00BB    10771               JMP     RDTRACK          ;| 16 READ FULL TRACK
018A7 FFFFF E0100    10772               JMP     COMREJ1          ;| 17
018A8 FFFFF E0100    10773               JMP     COMREJ1          ;| 18
018A9 FFFFF E0100    10775               JMP     COMREJ1          ;|$19 Format Defective is invalid
018AA FFFFF E0100    10779               JMP     COMREJ1          ;| 1A
```

```
 Addr              Line E.F. COMMAND DECODER

018AB FFFFF E0100 10780            JMP     COMREJ1          ;| 1B
018AC FFFFF E0100 10781            JMP     COMREJ1          ;| 1C
018AD FFFFF E0100 10782            JMP     COMREJ1          ;| 1D
018AE FFFFF E0100 10783            JMP     COMREJ1          ;| 1E
018AF FFFFF E0100 10784            JMP     COMREJ1          ;| 1F
018B0 FFFFF E0100 10785            JMP     COMREJ1          ;| 20
018B1 FFFFF E00A8 10786            JMP     PREPCYL          ;| 21 PREP CYLINDER
018B2 FFFFF E0100 10787            JMP     COMREJ1          ;| 22
018B3 FFFFF E0100 10788            JMP     COMREJ1          ;| 23
018B4 FFFFF E0CAF 10789            JMP     SENS15           ;| 24 READ AND RESET BUFFERED LOG
018B5 FFFFF E0100 10793            JMP     COMREJ1          ;| 25 'Clear Test Write': Invalid
018B6 FFFFF E00CD 10795            JMP     SPACE.RD         ;| 26 SPACE COUNT AND READ DATA
018B7 FFFFF E009D 10796            JMP     SEEK             ;| 27 SEEK AND SET SECTOR
018B8 FFFFF E0100 10797            JMP     COMREJ1          ;| 28
018B9 FFFFF E0100 10798            JMP     COMREJ1          ;| 29
018BA FFFFF E0100 10799            JMP     COMREJ1          ;| 2A
018BB FFFFF E009A 10800            JMP     RELSEEK          ;| 2B SECTOR POSITION RELATIVE SEEK
018BC FFFFF E0094 10801            JMP     RDRSTERR         ;| 2C READ AND RESET DRIVE ERROR COUNTERS
018BD FFFFF E0100 10802            JMP     COMREJ1          ;| 2D
018BE FFFFF E0100 10803            JMP     COMREJ1          ;| 2E
018BF FFFFF E0100 10804            JMP     COMREJ1          ;| 2F
018C0 FFFFF E0100 10805            JMP     COMREJ1          ;| 30 CDMF READ (Invalid Drive 0-30)
018C1 FFFFF E0100 10806            JMP     COMREJ1          ;| 31 CDMF WRITE (Inavlid Drive 0-30)
018C2 FFFFF E0100 10808            JMP     COMREJ1          ;|$32 Read Label not valid for CDC
018C3 FFFFF E0100 10812            JMP     COMREJ1          ;| 33
018C4 FFFFF E0CB6 10813            JMP     SENS18           ;| 34 READ CONFIGURATION
018C5 FFFFF E0100 10815            JMP     COMREJ1          ;|$35 Write H.A. & R0 Special is invalid
018C6 FFFFF E00C0 10819            JMP     RDSPECAL         ;| 36 READ HOME ADDRESS SPECIAL
018C7 FFFFF E0100 10820            JMP     COMREJ1          ;| 37
018C8 FFFFF E0100 10821            JMP     COMREJ1          ;| 38
018C9 FFFFF E0100 10822            JMP     COMREJ1          ;| 39
018CA FFFFF E0100 10823            JMP     COMREJ1          ;| 3A
018CB FFFFF E0100 10824            JMP     COMREJ1          ;| 3B
018CC FFFFF E0100 10825            JMP     COMREJ1          ;| 3C
018CD FFFFF E0100 10826            JMP     COMREJ1          ;| 3D
018CE FFFFF E00B4 10827            JMP     RDDATSPC         ;| 3E READ DATA SPECIAL
018CF FFFFF E0100 10828            JMP     COMREJ1          ;| 3F
                  10829                                     ;ENDCASE
                  10830    ;=================================================================
                  10831    TITLE HOST AND OCP MESSAGES
```

Addr Line HOST AND OCP MESSAGES

```
018D0             10833    ORG H#18D0
                  10834    ;
                  10835    ;        THIS ROUTINE SENDS MESSAGES TO THE OCP.
                  10836    ;        ENTER WITH THE MESSAGE NUMBER IN R0.
                  10837    ;Registers used:  R0-R2 & Q.
                  10838    ;
018D0 0F00F E0E24 10839    ISSMESS:  ADD DZC,Q,ZERO & JMP ISSCMD       ;SET MESSAGE FLAG & SEND MSG.
                  10934    ;
                  10935    ; E.C.  DISPLAY ROUTINE
                  10936    ;
                  10937    ;  ENTER WITH ADDRESS IN R0
                  10938    ;
018D1 15150 F07FF 10939    FRBDISP:  SBA DB,FOUTL,MSK2,R0 & NOP H#7FF  ;IF DISPLAY SELECT IS...
018D2 37160 A4FFF 10940              MOVE SMCST,ZERO & CRTS NAGB       ;LESS THAN 800, RETURN
                  10945    ;
018D3 B3700 8F000 10946              WRTE SCP,LDDISP3 & POP 0          ;CLEAR DISPLAY 0-11
018D4 B3600 F0000 10947              WRTE SCP,LDDISP2 & NOP 0          ;CLEAR DISPLAY 12-23
018D5 B3500 F0000 10948              WRTE SCP,LDDISP1 & NOP 0          ;CLEAR DISPLAY 24-35
018D6 B3400 5F800 10949              WRTE SCP,LDPRGD & JAR 0           ;CLEAR PROGRAM DISPLAY & RETURN
                  10950
                  10968    TITLE OCP DRIVE STATE ROUTINES
```

Addr Line OCP DRIVE STATE ROUTINES

```
                10970   ;   THIS ROUTINE SCANS THE BUFFER MEMORY FOR ANY OUTSTANDING REQUESTS.
                10971   ;   IF IT FINDS ANY, IT WILL PLACE THE STATE IN REQSTATE AND THE
                10972   ;   UNIT IN REQUNIT.
                10973   ;
                10974   ; EXIT PARAMETERS:      R0 = 0     NO REQUESTS AT THIS TIME
                10975   ;                       R0 = 1      FOUND A REQUEST
                10976   ;                       REQUNIT (SMB)  LOCATION OF UNIT WITH ACTIVE REQUEST
                10977   ;                                  0-31 = INDIVIDUAL UNIT
                10978   ;                                   H#FF = ALL UNITS
                10979   ;
                10980   ;                       REQSTATE (SMB) VALUE OF THE REQUESTED STATE
                10981   ;                                0  NO REQUEST
                10982   ;                                1  DOWN
                10983   ;                                2  CACHE
                10984   ;                                3  BYPASS
                10985   ;                                4  READ ONLY
                10986   ;                                5  WRITE THROUGH
                10987   ;REGISTERS USED: R0,R1,Q & IREG.
                10988   ;
018D7 3410E FFFFF 10989   REQSCAN:  ALDREG R0,IREG                    ;SAVE RETURN ADDRESS
                10990   ;
                10991   ; CHECK FOR "ALL REQUEST"
                10992   ;
018D8 B7000 F0020 10993             WRTE OCP,OCPADR & NOP D#32        ;SET ADDR FOR "ALL" LOCATION
018D9 34909 4F002 10994             ALDREGU R0,IREGH & PSHLD 2
018DA C7F10 887FF 10995             RD OCP,OCPDATA,R0 & CLPCNT EVCNTZ  ;GET "ALL" REQUEST
018DB 45350 F0007 10996             AND DB,B,MSK2,R0 & NOP H#7        ;ISOLATE REQUEST
018DC 433F1 330E3 10997             RESETR R1 & BOZ REQSC04           ;BRANCH IF NOT "ALL"
018DD 3410F F0262 10998             ALDRAM R0,REQSTATE                ;STORE THE REQUEST
018DE 37351 F00FF 10999             RDMSKB R1,H#FF                    ;SET "ALL"...
018DF 3411F F0261 11000             ALDRAM R1,REQUNIT                 ;AS THE REQUESTED UNIT
018E0 FFFFF 4F002 11001             PSHLD    2
018E1 B7100 88000 11002             WRTE OCP,OCPDATA & CLPCNT EVCNTZ,0 ;CLEAR "ALL" LOCATION
018E2 0F300 6F7FF 11003   REQRTN:   ADD DZC,B,ZERO,R0 & JIR            ;RETURN WITH R0=1
                11004   ;
                11005   ; SEARCH FOR INDIVIDUAL UNIT
                11006   ;
018E3 3710F F0262 11007   REQSC04:  RESETRAM REQSTATE                 ;CLEAR THE REQUESTED STATE
018E4 9701F 4F002 11008   REQSC01:  WRTL OCP,OCPADR,R1 & PSHLD 2      ;SET THE BUFFER ADDRESS
018E5 C7F10 887FF 11009             RD OCP,OCPDATA,R0 & CLPCNT EVCNTZ  ;GET THE DATA
018E6 45350 F0007 11010             MSKRBIT  R0,7                     ;ISOLATE REQUESTED STATE
018E7 0B3F1 338EB 11011             INCRB R1 & BNZ REQSC03            ;JUMP IF IT IS NOT ZERO
                11012   ;
018E8 65051 F0020 11013   REQSC02:  TSTRMSK R1,20                      ;CHECK IF LAST UNIT
018E9 433F0 637FF 11014             RESETR R0 & CJIR ALUZ             ;RETURN IF LAST UNIT
018EA FFFFF E18E4 11015             JMP REQSC01                       ;GO DO NEXT
018EB E7120 C0002 11016   REQSC03:  CTL OCP,SRESRV & LDCT 2           ;SET RESERVED
018EC 3410F F0262 11017             ALDRAM R0,REQSTATE                ;SAVE THE STATE
018ED 1411F F0261 11018             SBA AZ,FOUTL,R1,RAME & NOP REQUNIT ;SAVE THE UNIT
018EE C7F10 980EE 11019             RD OCP,OCPDATA,R0 & CJPCNT EVCNTZ,$ ;GET THE DATA BACK
018EF 45350 F0038 11020             MSKRBIT  R0,Q#70                  ;CLEAR REQUESTED & ISOLATE PRESENT
018F0 9710F 4F001 11021             WRTL OCP,OCPDATA,R0 & PSHLD 1     ;WRITE IT BACK
018F1 9710F 887FF 11022             WRTL OCP,OCPDATA,R0 & CLPCNT EVCNTZ
018F2 335F0 4F001 11023             OR       ZB,RSBZ,,R0 & PSHLD 1
```

Addr Line OCP DRIVE STATE ROUTINES

```
018F3 335F0 887FF 11024            RSHFTZ R0                        ;SHIFT PRESENT STATE RIGHT 3 BITS
018F4 3410F F0263 11025            ALDRAM R0,PRSSTATE               ;STORE PRESENT STATE
018F5 E7020 E18E2 11026            CTL OCP,RRESRV & JMP REQRTN      ;RESET RESERVED & RETURN
                  11027
                  11029    ;
                  11030    ;    THIS ROUTINE SETS THE PRESENT STATE FROM THE REQUESTED STATE AND UNIT
                  11031    ;
                  11032    ; ENTRY PARAMETERS:     REQSTATE (SMB)
                  11033    ;                          =0  NO REQUESTS
                  11034    ;                          =1  DOWN
                  11035    ;                          =2  CACHE
                  11036    ;                          =3  BYPASS
                  11037    ;                          =4  READ ONLY
                  11038    ;                          =5  WRITE THROUGH
                  11039    ;
                  11040    ;                       REQUNIT (SMB)
                  11041    ;                          00-1F  SPECIFY UNIQUE UNIT#
                  11042    ;                          FF     CHANGE ALL UNITS
                  11043    ;REGISTERS USED: R0,R1,Q & IREG.
                  11044    ;
018F6 3410E FFFFF 11045    SETPREZ:  ALDREG R0,IREG                    ;SAVE RETURN
018F7 34909 C0000 11046              ALDREGU R0,IREGH & LDCT 0
018F8 373F1 F0262 11047              RDRAMB R1,REQSTATE                ;GET THE REQUESTED STATE
018F9 373F0 F0261 11048              RDRAMB R0,REQUNIT                 ;GET THE REQUESTED UNIT
018FA 65050 F00FF 11049              TSTRMSK R0,H#FF                   ;CHECK FOR ALL UNITS
018FB FFFFF 338FD 11050              BNZ SETPREZA                      ;JUMP IF NOT ALL UNITS
018FC 433F0 C0020 11051              RESETR R0 & LDCT 32               ;SET UNIT=0 & ITERATION COUNT
018FD 01711 FFFFF 11052    SETPREZA: ADD AB,LSBZ,R1,R1                 ;SHIFT REQUESTED INTO PRESENT
018FE 337F1 FFFFF 11053              OR ZB,LSBZ,,R1
018FF 45351 F0038 11054              MSKRBIT  R1,38                    ;MASK OUT JUNK
01900 9700F FFFFF 11055    SETPREZB: WRTL OCP,OCPADR,R0                ;SET THE OCP ADDRESS
01901 E7020 FFFFF 11056              CTL OCP,RRESRV                    ;CLEAR RESERVED
01902 9711F FFFFF 11057              WRTL OCP,OCPDATA,R1               ;WRITE THE DATA
01903 9711F FFFFF 11058              WRTL     OCP,OCPDATA,R1
01904 9711F FFFFF 11059              WRTL     OCP,OCPDATA,R1
01905 0B3F0 98100 11060              INCRB R0 & CJPCNT EVCNTZ,SETPREZB
01906 FFFFF 6F7FF 11061              JIR
                  11062
                  11064    TITLE CACHE SEEK PROCESSING
```

Addr Line CACHE SEEK PROCESSING

```
                11092   ;THIS IS THE SEEK ROUTINE FOR CACHE.
                11093   ;FIRST IT CHECKS THE DRIVE STATE AND A FEW FLAGS.  THEN IT BUILDS THE TRACK
                11094   ;ADDRESS FOR THE E.C. SEARCH, THEN SEARCHES THE E.C.  IF THE TRACK IS IN CACHE,
                11095   ;A STATUS OF CHANNEL END AND DEVICE END WITH A STATUS ACTION OF 5 IS
                11096   ;SENT TO THE CHANNEL.  IF THE TRACK IS IN CACHE AND THE SEEK IS AN
                11097   ;IMBEDDED ONE, A RETURN TO THE STACK IS DONE.
                11098   ;
01907 371F0 F0302 11099   RATSEEK:  MOVE     RAME,ZERO & NOP SEEKONL   ;Seek Only flag set?
01908 3710F A3B02 11100             MOVE     ZERO,RAME & CRTS NALUZ,SEEKONL;Yes, reset it & return
01909 373F4 F0204 11101             RDRAMB   R4,FLAGSAV                ;Get status flags
0190A 3710F F0255 11102             RESETRAM ECCSTAT                   ;Reset ECCSTAT reg.
0190B 3710F F0242 11103             RESETRAM  CASHSTAT                 ;CLEAR CACHE STATUS FLAGS
                11104   ;
                11105               JSR.P LOGICAL                      ;ISOLATE LOGICAL DRIVE NO.
0190C E0400 1F1F9 11105 +           SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
0190D 321FF F0202 11106             QLDRAM    NEWDRV                   ;SAVE DRIVE NO. FROM E.F.
0190E 3415F F0200 11107             ALDRAM    R5,R5RAT                 ;SAVE R5 LOWER
0190F 34312 FFFFF 11108             RTOR      R1,R2                    ;MOVE HEAD # TO R2
01910 373F1 F0263 11109             RDRAMB    R1,PRSSTATE              ;GET PRESENT STATE
01911 FFFFF 3311C 11110             BOZ       RATSEEK3                 ;IF ZERO - ERROR
01912 65051 F0001 11111   RATSEEK0: TSTRMSK   R1,1                     ;DRIVE STATUS = 'DOWN'?
                11112               CJPOP.P ALUZ,SEL22                 ;Yes, Intervention Required
01913 E0401 B32F3 11112 +           SETPGE SEL22_SHR_H#B & CJPOPP ALUZ,SEL22
01914 65051 F0003 11113             TSTRMSK R1,3                       ;IS STATUS 'BYPASS'?
01915 FFFFF 33917 11115             BNZ       $+2                      ;$If not, continue
                11119   RATSEEK1: EQU $
01916 34321 AF7FF 11123             RTOR R2,R1 & RTS                   ;Head # back to R1 & return
                11124   ;
01917 65051 F0004 11125             TSTRMSK   R1,4                     ;IS STATUS 'READ ONLY'?
01918 FFFFF 3391C 11127             BNZ      RATSEEK3                  ;$No, try to get drive state again
01919 433F3 1F1C3 11131             RESETR    R3 & JSR CHK4WRIT        ;SEE IF IT IS A WRITE FUNCTION
                11132               CJPOP.P NALUZ,RDONLY               ;Yes, send 'Read Only' status
0191A E0400 B3914 11132 +           SETPGE RDONLY_SHR_H#B & CJPOPP NALUZ,RDONLY
0191B FFFFF E1916 11134             JMP       RATSEEK1                 ;$Otherwise, return
0191C 373F1 F0202 11178   RATSEEK3: RDRAMB R1,NEWDRV                   ;GET THE DRIVE NUMBER
                11179               JSR.P DRIVSTAT                     ;ATTEMPT TO GET GOOD DRIVE STATE
0191D E0404 1F000 11179 +           SETPGE DRIVSTAT_SHR_H#B & JSRP DRIVSTAT
```

Addr Line CACHE SEEK PROCESSING

```
0191E 333F8 33110 11180          OR ZB,B,,R8 & BOZ $-1          ;IF OCP NOT SELECTED, TRY AGAIN
0191F FFFFF 33122 11182          BOZ     $+3                    ;$If drive state = 0, force BYPASS
01920 2D148 F0004 11183          SUBB DB,FOUTL,MSK1,R8 & NOP H#4 ;$Drive state > 4?
01921 FFFFF 35923 11184          CJMP    NALB,$+2               ;$If not, jump
01922 37358 F0003 11185          RDMSKB  R8,3                   ;$Otherwise, drive state = BYPASS
01923 3418F F0263 11191          ALDRAM R8,PRSSTATE             ;SAVE DRIVE STATE
01924 34381 E1912 11192          RTOR R8,R1 & JMP RATSEEK0      ;CONTINUE
```

Addr Line CACHE SEEK PROCESSING

```
                11697   ;    THIS ROUTINE WILL CHECK IF THIS SYSTEM HAS TWO DRIVES PER UNIT.
                11698   ;IF IT DOES, THE UNIT NO. IN R13 WILL BE DOUBLED.
                11699   ;
                11700   DOUBLE:   EQU $
01925 371F0 F02F4 11702           MOVE     RAME,ZERO & NOP TWODRIV   ;$Two drives any unit?
01926 FFFFF A37FF 11707           CRTS ALUZ                          ;IF NOT, RETURN
01927 337FD AF7FF 11708           OR ZB,LSBZ,,R13 & RTS              ;ELSE, DOUBLE THE UNIT NO.
                11709
```

Addr Line CACHE SEEK PROCESSING

```
                11901   ;
                11902   ; ---- SEEK ERROR PROCESSING ----
                11903   ;
                11904   ;THE SEEK ERROR PROCESSING ROUTINES CONSIST OF 4 SEPARATE ROUTINES.
                11905   ;THERE ARE TWO FOR PROCESSING ERRORS ON THE "READ" DRIVE, AND TWO FOR
                11906   ;PROCESSING ERRORS ON THE "WRITE" DRIVE (WHEN IT IS DIFFERENT FROM THE
                11907   ;"READ" DRIVE).  EACH PAIR CONSISTS OF A ROUTINE TO PROCESS A SEEK ERROR
                11908   ;AND A ROUTINE TO PROCESS A TIMEOUT.  THE ONLY DIFFERENCE BETWEEN THE
                11909   ;SEEK ERROR AND THE TIMEOUT IS THAT FOR A SEEK ERROR, SEEK ERROR LOGGING
                11910   ;IS DONE. THE PROCESSING ON THE READ DRIVE IS DONE BY THE STANDARD
                11911   ;5046 SEEK ERROR PROCESSING CODE SO THIS CODE IS SIMPLY AN INTERFACE
                11912   ;TO IT.  THE PROCESSING ON THE WRITE DRIVE IS ACTUALLY DONE HERE.
                11913   ;THE PROCESSING CONSISTS OF A RECALIBRATE FOLLOWED BY A RE-SEEK TO
                11914   ;THE CORRECT CYLINDER. IF THE ERROR PERSISTS AND THE CALL IS FROM
                11915   ;SEEK CODE, THE E.C. IS RESTORED TO THE OLD DATA AND ANOTHER TRACK IS
                11916   ;PUSHED OFF.  IF THE CALL IS FROM WRITE, RETURN IS TO THE ADDRESS REGISTER.
                11917   ;THE FOUR ROUTINES ARE AS FOLLOWS:
                11918   ;SEEKERR1 - READ DRIVE SEEK ERROR
                11919   ;SEEKERRT - READ DRIVE TIMEOUT
                11920   ;SEEKERR2 - WRITE DRIVE SEEK ERROR
                11921   ;SEEKERR3 - WRITE DRIVE TIMEOUT
                11922   ;
01928 FFFFF 1F12F 11923   SEEKERR1: JSR ERRSAV                      ;SAVE SOME REGISTERS
                11924             JSR.P SEEK30                    ;PROCESS FAULT &/OR SEEK ERROR
01929 E0400 1F18B 11924 +           SETPGE SEEK30_SHR_H#B & JSRP SEEK30
0192A FFFFF 1F172 11925   SEEKERRA: JSR GETREGS                     ;RESTORE SAVED REGISTERS
0192B 353FF AF239 11930             OR DB,B,RAME,R15 & RTS SEEKFLG  ;RESTORE SEEK ONLY FLAG & RETURN
                11932   ;
0192C FFFFF 1F12F 11933   SEEKERRT: JSR ERRSAV
                11934             JSR.P SEEK37                    ;PROCESS TIMEOUT
0192D E0400 1F1CC 11934 +           SETPGE SEEK37_SHR_H#B & JSRP SEEK37
0192E FFFFF E192A 11935             JMP SEEKERRA                    ;RETURN
                11953   ;
0192F FFFFF 1F16D 11954   ERRSAV:   JSR SAVREGS                     ;SAVE SOME REGISTERS
01930 4505F F0080 11955             TSTRBIT R15,80                  ;ISOLATE SEEK ONLY FLAG...
01931 321FF F0239 11959             QLDRAM SEEKFLG                  ;AND SAVE IT
01932 5534F AF080 11960             MASK DB,B,MSK1,R15 & RTS H#80   ;RESET IT FOR ERROR PROCESSING
                11961   ;
                12002   TITLE CACHE SEEK PROCESSING SUBROUTINES
```

Addr Line CACHE SEEK PROCESSING SUBROUTINES

```
                12004
                12005   ;THIS SUBROUTINE CHECKS THE AVAILABILITY OF A DRIVE.
                12006   ;THE PHYSICAL UNIT # IS EXPECTED IN R13. THE ROUTINE IS CALLED USING
                12007   ;THE STACK. RETURN IS TO $+1. IF THE DEVICE IS
                12008   ;UNAVAILABLE, I.E. NOT (READY AND SELECTED AND ON-CYL) OR BUSY
                12009   ;OR FAULT, THE ALU WILL BE NON-ZERO.
                12010   ;AT EXIT R0 CONTAINS DEVICE STATUS.
                12011   ;THIS ROUTINE USES: R0,R4,R8,R10,Q & IREG.
                12012   ;
01933 5535E F0800 12013   CHKAVALA: RSTRBIT R14,800                   ;RESET SECOND DRIVE FLAG
01934 3491E FFFFF 12014           ALDREGU R1,IREG                   ;CYLINDER TO IREG...
01935 373E0 FFFFF 12015           RDREGB IREG,R0                    ;AND THEN TO R0
01936 FFFFF 1F193 12019   CHKAVALB: JSR CHK2ND                       ;CHECK FOR SECOND DRIVE
01937 4515E F0800 12021           AND DB,FOUTL,MSK2,R14 & NOP H#800 ;SECOND DRIVE FLAG SET?
01938 FFFFF 3394D 12022           BNZ CHKAVAL0                      ;YES, DRIVE IS ALREADY SELECTED
                12023   ;
01939 B2200 F0000 12024   CHKAVAL:  WRTE HSSMA,LDDRBO & NOP 0         ;CLEAR DATA BUS OUT REG.
0193A E100B FFFFF 12025           CTL       HSSM,RSTUSEL            ;DE-SELECT DRIVE
0193B 329FF F02EE 12026           QLDRAMU SAVQU                     ;SAVE Q-UPPER AND...
0193C 321FF F02EF 12027           QLDRAM SAVQL                      ;Q-LOWER
0193D 4505D F0030 12028           TSTRBIT R13,30                    ;CARD CAGE SELECT (CHALLNGR)...
0193E 323F4 FFFFF 12029           QTOR R4                           ;BITS INTO R4
0193F 341DF F000D 12030           ALDRAM R13,SEN113                 ;SAVE DRIVE NO. FOR SENSE
01940 91EDF 4F00A 12031           WRTL HSSM,LDDRUSEL,R13 & PSHLD 10 ;SET UNIT SELECT REGISTER
01941 C1040 FFFFF 12032           RD HSSM,RDDRST1,R0
01942 C1040 B8145 12033           RD HSSM,RDDRST1,R0 & CJPOP EVCNTZ,$+3 ;READ DRIVE STATUS
01943 E1007 83FFF 12034           CTL HSSM,SETUSEL & CLPCNT NALUZ   ;IF NONE, JUMP BACK; SELECT DRIVE
01944 91E4F FFFFF 12035           WRTL HSSM,LDDRUSEL,R4             ;CARD CAGE SELECT ONLY
01945 C1040 98145 12036           RD HSSM,RDDRST1,R0 & CJPCNT EVCNTZ,$ ;DELAY FOR GOOD STATUS
01946 45150 F0040 12037           TSTBIT R0,BUSY                    ;DRIVE BUSY?
01947 FFFFF 33953 12038           BNZ CHKAVAL2                      ;IF IT IS, RETURN
                12039   ;
01948 45050 F0008 12040           TSTRBIT R0,FAULT                  ;DRIVE FAULT?
01949 FFFFF 13968 12041           CJSR NALUZ,CHKFALT                ;IF YES, ATTEMPT TO RESET IT
0194A 45050 F0021 12042           TSTRBIT R0,SELECT+READY           ;ISOLATE SELECTED & READY
0194B 6605F F0021 12043           TSTQMSK SELECT+READY              ;SELECTED & READY BOTH SET?
0194C FFFFF 33953 12044           BNZ CHKAVAL2                      ;IF NOT, RETURN
                12045   ;
                12046   CHKAVAL0: JSR.P FLAGLOC                     ;FLAG LOC. THIS DRIVE
0194D E0401 1F342 12046 +         SETPGE FLAGLOC_SHR_H#B & JSRP FLAGLOC
0194E 370DF FFFFF 12047           RDREGQ RAMI                       ;GET THE FLAGS
0194F FFFFF 33159 12048           BOZ CHKAVAL3                      ;IF NO FLAGS, JUMP
01950 661F0 F03E5 12049           XOR DQ,FOUTL,RAME,ZERO & NOP LASTFLG ;THIS FLAG SAME AS LAST FLAG?
01951 C1040 3395B 12050           RD HSSM,RDDRST1,R0 & BNZ CHKAVAL4 ;IF NOT, JUMP
01952 C1040 FFFFF 12051   CHKAVAL1: RD HSSM,RDDRST1,R0
01953 55040 F0010 12052   CHKAVAL2: MASK DB,Q,MSK1,R0 & NOP WRITPROT  ;DON'T CHECK WRITE PROTECT
01954 6615E F00A3 12053           XOR DQ,FOUTL,MSK2,IREG & NOP ONCYL+READY+SELECT+SKEND ;GOOD STATUS?
01955 373F4 F02EE 12054           RDRAMB R4,SAVQU
01956 D0FF4 FFFFF 12055           RDXFR SMC,,R4
01957 350F4 F02EF 12056           OR DB,Q,RAME,R4 & NOP SAVQL       ;RESTORE Q-REG...
```

Addr Line CACHE SEEK PROCESSING SUBROUTINES

```
01958 371EE AF7FF 12057             MOVE IREG,IREG & RTS            ;AND RETURN
                  12058     ;
01959 45050 F0002 12059     CHKAVAL3: TSTRBIT R0,ONCYL              ;IS DRIVE ON CYLINDER?
0195A FFFFF 33153 12060             BOZ CHKAVAL2                    ;IF NOT, WE CAN'T READ A HEADER
0195B FFFFF 1F16E 12061     CHKAVAL4: JSR SAVREGS+1                 ;SAVE R1, R2 & R3
                  12062             SVWD R5,SAVR5U                  ;SAVE R5
0195C 3495F F02EC 12062 +           ALDRAMU  R5,SAVR5U              ;SAVE 12 MSBS
0195D 3415F F02ED 12062 +           ALDRAM   R5,SAVR5U+1            ;SAVE 12 LSBS
0195E 3416F F02F2 12063             ALDRAM R6,SAVR6                 ;SAVE R6
0195F 3535E F0020 12064             SETRBIT R14,20                  ;SET SPECIAL FLAG
                  12065             JSR.P SEL10A                    ;SET UP CONSTANTS & LIMITS
01960 E0401 1F1E5 12065 +           SETPGE SEL10A_SHR_H#B & JSRP SEL10A
                  12066     ;
01961 FFFFF 1F173 12067             JSR GETREGS+1                   ;RESTORE R1, R2 & R3
                  12068             LDWD R5,SAVR5U                  ;Restore saved R5
01962 373F5 F02EC 12068 +           RDRAMB   R5,SAVR5U              ;GET 12 MSBS
01963 D0FF5 FFFFF 12068 +           RDXFR    SMC,,R5                ;PUT IN UPPER REGISTER
01964 353F5 F02ED 12068 +           OR DB,B,RAME,R5 & NOP SAVR5U+1  ;OR IN 12 LSBS
01965 373F6 F02F2 12069             RDRAMB R6,SAVR6                 ;RESTORE SAVED R6
01966 5535E F0020 12070             RSTRBIT R14,20                  ;RESET SPECIAL FLAG
01967 C1040 E1952 12071             RD HSSM,RDDRST1,R0 & JMP CHKAVAL1
                  12072     ;
01968 B2200 F0010 12073     CHKFALT:  WRTE HSSMA,LDDRBU & NOP H#10  ;BUS OUT FOR FAULT RESET
01969 E100A 4F00A 12074             CTL HSSM,SETAG3 & PSHLD 10      ;ATTEMPT TO RESET THE FAULT
0196A C1040 887FF 12075             RD HSSM,RDDRST1,R0 & CLPCNT EVCNTZ ;DELAY FOR RESET TO WORK
0196B E1016 FFFFF 12076             CTL HSSM,RSTAG3                 ;RESET TAG 3
0196C B2200 AF000 12077             WRTE HSSMA,LDDRBU & RTS 0       ;CLEAR BUS OUT & RETURN
                  12078
0196D 321FF F02EF 12079     SAVREGS:  QLDRAM SAVQL                  ;SAVE THE Q-REGISTER
                  12080             SVWD R1,SAVR1U                  ;SAVE R1
0196E 3491F F02E8 12080 +           ALDRAMU  R1,SAVR1U              ;SAVE 12 MSBS
0196F 3411F F02E9 12080 +           ALDRAM   R1,SAVR1U+1            ;SAVE 12 LSBS
01970 3412F F02EA 12081             ALDRAM R2,SAVR2                 ;SAVE R2 AND...
01971 3413F AF2E8 12082             ALDREG R3,RAME & RTS SAVR3      ;R3; RETURN
                  12083     ;
01972 370FF F02EF 12084     GETREGS:  RDRAMQ SAVQL                  ;RESTORE Q-REGISTER
                  12085             LDWD R1,SAVR1U                  ;RESTORE SAVED R1
01973 373F1 F02E8 12085 +           RDRAMB   R1,SAVR1U              ;GET 12 MSBS
01974 D0FF1 FFFFF 12085 +           RDXFR    SMC,,R1                ;PUT IN UPPER REGISTER
01975 353F1 F02E9 12085 +           OR DB,B,RAME,R1 & NOP SAVR1U+1  ;OR IN 12 LSBS
01976 373F2 F02EA 12086             RDRAMB R2,SAVR2                 ;RESTORE R2 AND...
01977 373F3 AF2E8 12087             RDREGB RAME,R3 & RTS SAVR3      ;R3; RETURN
```

Addr Line CACHE SEEK PROCESSING SUBROUTINES

```
                12100   ;
                12101   ;THIS ROUTINE WAITS FOR THE READ DRIVE TO COMPLETE SEEKING
                12102   ;
01978 0F320 E197C 12103   RDSKWAIT: CALL R0, WAITSEEK               ;CALL THE WAIT HANDLER
01979 FFFFF E1928 12104            JMP SEEKERR1                    ;PROCESS SEEK ERROR
0197A FFFFF E192C 12105            JMP SEEKERRT                    ;PROCESS TIME-OUT
0197B FFFFF AF7FF 12106            RTS                             ;RETURN IF SEEK COMPLETES NORMALLY
                12107   ;
                12108   ;SUBROUTINE WAITSEEK WILL WAIT FOR SEEK COMPLETION ON A DRIVE. THE PHYSICAL
                12109   ;DRIVE NO. IS IN R13. THE ROUTINE IS CALLED ON R0 & USES R0,R4,R8,R10 & IREG.
                12110   ;RETURNS TO $+1 IF SEEK ERROR, TO $+2 IF 2-SEC. TIME OUT, OR TO
                12111   ;$+3 IF NORMAL COMPLETION.
                12112   ;
0197C 3490F F025B 12113   WAITSEEK: ALDRAMU  R0, WAITRETU          ;SAVE RET ADD UPPER...
0197D 3410F F02F1 12114            ALDRAM R0,WAITRETL              ;AND LOWER
0197E FFFFF 1F139 12115            JSR CHKAVAL                     ;SELECT THE DRIVE
0197F FFFFF 3318D 12116            BOZ WAITRET+1                   ;IF ON CYL., RETURN NOW
01980 371FE F02F1 12117            MOVE RAME,IREG & NOP WAITRETL   ;RETURN LSB'S TO IREG
01981 45150 F0020 12118            TSTBIT R0,SELECT                ;SELECTED?
01982 FFFFF 331BF 12119            BOZ       RET1                  ;NO - ERROR RETURN $+1
01983 37354 F00AE 12120            RDMSKB R4,H#AE
01984 D0FF4 FFFFF 12121            RDXFR SNC,,R4
01985 35354 F062D 12122            SETRBIT R4,62D                  ;2 SECOND TIMER TO R4
01986 3418F F02F2 12123            ALDRAM R8,SAVR6                 ;TEMPORARILY SAVE R8
                12124   ;
01987 FFFFF 1F1B2 12125            JSR       CHKDRV                ;DRIVE ON CYLINDER?
01988 133F4 3318C 12126            DECRB R4  & BOZ WAITRET         ;DEC TIMER; IF ON CYL., RETURN $+3
01989 FFFFF 33987 12127            BNZ       $-2                   ;IF TIMER #EXPIRED, TRY AGAIN
                12128   ;
0198A 373F8 F02F2 12129            RDRAMB R8,SAVR6                 ;TIMER EXPIRED; RESTORE R8...
0198B FFFFF E198F 12130            JMP       RET2                  ;AND RETURN $+2
                12131   ;
0198C 373F8 F02F2 12132   WAITRET: RDRAMB R8,SAVR6                 ;RESTORE R8
0198D 371FE F02F1 12133            MOVE RAME,IREG & NOP WAITRETL   ;RETURN LSB'S TO IREG
                12191   ;
0198E 0F1EE FFFFF 12192   RET3:     INCRD     IREG,IREG             ;BUMP IREG FOR NOT ON-CYL RETURN
                12193   ;
0198F 0F1EE FFFFF 12194   RET2:     INCRD     IREG,IREG             ;BUMP IREG FOR ON-CYL OR INITIATED
01990 371F9 F025B 12195            MOVE RAME, IREGH & NOP WAITRETU ;GET RETURN ADDR UPPER
01991 3710F 6F02C 12196            MOVE ZERO,RAME & JIR SKRETRY    ;RESET ERR CNT & RETURN
```

Addr Line CACHE SEEK PROCESSING SUBROUTINES

```
                12198   ;    FIRST, THIS ROUTINE WILL CHECK IF THE SYSTEM HAS ANY TWO-DRIVE UNITS.
                12199   ;IF IT DOES, IT WILL CHECK IF THIS IS A TWO-DRIVE UNIT.  IF IT IS, IT WILL
                12200   ;CHECK IF THE CYLINDER (R0) IS GREATER THAN THE MAX. CYL. OF THE FIRST DRIVE.
                12201   ;IF IT IS, THE UNIT NO. WILL BE INCREMENTED AND THE SECOND DRIVE FLAG WILL BE
                12202   ;SET.  IF NOT, THE CYL. NO. WILL BE UN-CHANGED, THE UNIT NO. WILL BE
                12203   ;DECREMENTED AND THE SECOND DRIVE FLAG WILL BE CLEARED.  IF THERE IS A NEED
                12204   ;TO GO TO THE OTHER DRIVE, THE CURRENT DRIVE WILL BE DE-SELECTED AND THE OTHER
                12205   ;SELECTED.
                12206   ;UPON ENTRY:  R0 = CURRENT CYLINDER
                12207   ;             R13= PHYSICAL DRIVE NO.
                12208   ;UPON EXIT:   R0 = SEEK CYLINDER
                12209   ;REGISTERS USED: R0,R4,R8,Q & 1REG.
                12210   ;
01992 343B0 FFFFF 12211   CHK2NDA:  RTOR R11,R0                       ;MOVE CYLINDER TO R0
                12212   CHK2ND:   EQU $
01993 370FF F02F4 12214             RDREGQ   RAME & NOP TWODRIV       ;$Any two-drive units?
01994 FFFFF A37FF 12219             CRTS ALUZ                         ;IF NOT, RETURN
                12220               JSR.P FLAGLOC                     ;LOC. OF DRIVE FLAGS
01995 E0401 1F342 12220 +           SETPGE FLAGLOC_SHR_H#B & JSRP FLAGLOC
01996 370DF FFFFF 12221             RDREGQ RAM1                       ;GET THEM
01997 FFFFF 3319A 12222             BOZ $+3                           ;IF NONE, ASSUME TWO DRIVE UNIT
01998 46150 F0400 12223             TESTQ TWOFER                      ;IS THIS A TWO-DRIVE UNIT?
01999 FFFFF 33180 12224             BOZ CHK2ND3                       ;IF NOT, JUMP
                12225   ;
0199A 3410F F039A 12226             ALDRAN R0,TEMPCYL                 ;TEMPORARILY SAVE CYLINDER
0199B 661F0 F03E5 12227             XOR DQ,FOUTL,RAME,ZERO & NOP LASTFLG ;THIS FLAG SAME AS LAST FLAG?
0199C FFFFF 1395B 12228             CJSR NALUZ,CHKAVAL4               ;IF NOT, COMPUTE NEW CONSTANTS
0199D 373F0 F039A 12229             RDRANB R0,TEMPCYL                 ;RESTORE SAVED CYLINDER
0199E 373F4 F02D4 12230             RDRANB R4,MAXCYLD                 ;GET MAX. DRIVE CYLINDER
0199F 335F4 FFFFF 12236             OR ZB,RSBZ,,R4                    ;1/2 MAX. DRIVE CYL. TO R4
019A0 29304 FFFFF 12237             SUBB AB,B,R0,R4                   ;CUR. CYL.> 1/2 MAX. CYL?
019A1 FFFFF 349AC 12238             CJMP NAGB,CHK2ND2                 ;IF NOT, JUMP
019A2 3535D F0001 12239             SETRBIT R13,1                     ;INCR. DRIVE NO.
019A3 4505E F0800 12240             TSTRBIT R14,800                   ;SECOND DRIVE FLAG ALREADY SET?
019A4 14340 A3FFF 12241             SBA AZ,B,R4,R0 & CRTS NALUZ       ;YES, RETURN; SEEK CYL. TO R0
                12242   ;
019A5 3535E F0800 12243             SETRBIT R14,800                   ;SET SECOND DRIVE FLAG
                12244   ;
                12248   CHK2ND1:  JSR.P RELDRIV                     ;RELEASE THIS DRIVE
019A6 E0401 1F60F 12248 +           SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
                12250   ;
019A7 3404F 1F139 12251             RTOQ R4 & JSR CHKAVAL             ;SELECT THE NEXT DRIVE
019A8 123F0 A37FF 12252             SBA ZQ,B,,R0 & CRTS ALUZ          ;IF ON CYL. & READY, RETURN
                12253   ;
```

Addr Line CACHE SEEK PROCESSING SUBROUTINES

```
019A9 4505E F0200 12254            TSTRBIT R14,200                  ;ALT. TRACK FLAG?
019AA 3400F 13978 12256            RTOQ R0 & CJSR NALUZ,RDSKWAIT    ;$Yes, wait for the drive
019AB 323F0 AF7FF 12264            QTOR R0 & RTS                    ;SEEK CYL. TO R0 & RETURN
                  12265    ;
019AC 5535D F0001 12266    CHK2ND2: RSTRBIT R13,1                    ;DECR. DRIVE NO.
019AD 4505E F0800 12267            TSTRBIT R14,800                  ;SECOND DRIVE FLAG SET?
019AE 603EE A37FF 12268            XOR AQ,B,R14,R14 & CRTS ALUZ     ;YES, RESET IT; NO, RETURN
019AF 0C304 E19A6 12269            ADD AZC,B,R0,R4 & JMP CHK2ND1    ;INCR. CYL.+1 & SELECT 1ST DRIVE
                  12270    ;
019B0 5535E F0800 12271    CHK2ND3: RSTRBIT R14,800                  ;RESET SECOND DRIVE FLAG
019B1 FFFFF AF7FF 12272            RTS                              ;RETURN
                  12282    ;
                  12283    ;ROUTINE TO TEST DRIVE AVAILABILITY. CALLED ON STACK, RETURNS
                  12284    ;TO IREG & POPS STACK IF NON RESETABLE FAULT. RETURNS TO STACK
                  12285    ;OTHERWISE. IF R0=0 THEN DRIVE IS AVAIL. & ON CYLINDER, ELSE R0 WILL NOT = 0.
                  12286    ;REGISTERS USED:  R0 & R8.
                  12287    ;
019B2 C1040 FFFFF 12288    CHKDRV:  RD HSSM,RDDRST1,R0
019B3 C1040 FFFFF 12289            RD HSSM,RDDRST1,R0               ;READ DRIVE STATUS
019B4 45150 F0008 12290            TSTBIT R0,FAULT                  ;DRIVE FAULT?
019B5 FFFFF 13968 12291            CJSR NALUZ,CHKFALT               ;IF YES, ATTEMPT TO RESET IT
                  12292    ;
019B6 45150 F000C 12293            TSTBIT R0,SEEKERR+FAULT          ;SEEK ERROR OR FAULT
019B7 C1040 B39BF 12294            RD HSSM,RDDRST1,R0 & CJPOP NALUZ,RET1 ;IF EITHER, ERROR RETURN
019B8 C1040 C0009 12295            RD HSSM,RDDRST1,R0 & LDCT 9
019B9 45150 F0041 12296            TSTBIT R0,READY+BUSY             ;READY OR BUSY
019BA 34308 339C1 12297            RTOR R0,R8 & BNZ RET1A           ;IF EITHER, NORMAL RETURN
                  12298    ;
019BB C1040 9B1BB 12299            RD HSSM,RDDRST1,R0 & CJPCNT EVCNTZ,$ ;DELAY 2 µsec. FOR GOOD STATUS
019BC 45150 F0041 12300            TSTBIT R0,READY+BUSY             ;CK. AGAIN FOR READY OR BUSY
019BD 34308 339C1 12301            RTOR R0,R8 & BNZ RET1A           ;IF EITHER, NORMAL RETURN
019BE FFFFF 8F7FF 12302            POP                              ;NO - POP THE STACK
                  12303    ;
019BF 371F9 F025B 12304    RET1:    MOVE RAME, IREGH & NOP WAITRETU  ;SET UPPER RETURN FIELD AND...
019C0 FFFFF 6F7FF 12305            JIR                              ;RETURN IN ERROR
                  12306    ;
019C1 55358 F0010 12307    RET1A:   RSTRBIT  R8,WRITPROT             ;DON'T CHECK WRITE PTOTECT
019C2 65348 AF0A3 12308            XOR DB,B,MSK1,R8 & RTS ONCYL+READY+SELECT+SKEND
                  12309    ;                                         CHECK GOOD DRIVE STATUS & RETURN
                  12310
                  12311    ;ROUTINE TO CHECK IF THE CURRENT FUNCTION IS A WRITE.
                  12312    ;IT IS IF THE SEEK-ONLY FLAG IS NOT SET OR THE WRITE-OP. FLAG IN R5 IS SET.
                  12313    ;IF THE FUNCTON IS A WRITE, THE WRITE-OP. FLAG WILL BE SET IN R3.
                  12314    ;THE ROUTINE IS CALLED ON THE STACK AND USES Q.
                  12315    ;
019C3 4505F F0080 12316    CHK4WRIT: TSTRBIT  R15,80                 ;IS SEEK-ONLY FLAG SET?
019C4 420FF A3FFF 12317            RESETQ   & CRTS NALUZ            ;IF YES, IT CANT BE A WRITE
019C5 370FF F0200 12318            RDRAMQ   R5RAT                   ;PICK UP SAVED R5
019C6 4605F F0800 12319            TSTQBIT  800                     ;WRITE FLAG SET?
019C7 320FF A37FF 12320            OR       ZQ,Q & CRTS ALUZ        ;IF NOT IT ISNT A WRITE
```

```
 Addr            Line CACHE SEEK PROCESSING SUBROUTINES

019C8 35343 AF001 12321            OR DB,B,MSK1,R3 & RTS WRITEOP     ;SET WRITE-OP AND RETURN
                  12362
                  12363   TITLE CLOCK AND OCP INTERRUPT PROCESSING
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
                12365
                12366   ; THIS CODE HANDLES THE WRITES & E.C. UPDATES REQUIRED BY DAYCLOCK
                12367   ; INTERRUPTS AND OCP DIRECTED DRIVE STATE CHANGES.
                12368   ; THE E.C. IS SEARCHED FOR ENTRIES WITH THE WRITE-UP FLAG SET. WHEN ONE
                12369   ; IS FOUND - IF THERE WAS NOT AN INTERVAL TYPE TIMER INTERRUPT OR IF THE
                12370   ; WRITE UPDATE FLAG IS SET, THE TRACK IS WRITTEN OUT. IF IT WAS AN INTERVAL
                12371   ; TIMER INTERRUPT AND THE WRITE UPDATE FLAG IS NOT SET THEN IT WILL BE SET.
                12372   ; THIS IS REPEATED FOR THE WHOLE E.C. IF THE REQUESTED STATE IS BYPASS OR DOWN,
                12373   ; THE E.C. TRACKS ARE INVALIDATED.
                12374   ;
                12375   ; THE SUBROUTINE IS CALLED VIA THE STACK.
                12376   ; REGISTER R6 HAS FLAGS.
                12377   ; BIT 0 MEANS INTERVAL INTERRUPT.
                12378   ; BIT 1 MEANS THAT THE FUNCTION IS A DRIVE STATE CHANGE.
                12379   ; BIT 2 MEANS REQUESTED STATE IS BYPASS OR DOWN.
                12380   ; REGISTER R7 IS USED TO CONTAIN THE UNIT # FOR STATE CHANGES.  IT
                12381   ; WILL CONTAIN HEX FF IF THE STATE CHANGE WAS AN 'ALL' FUNCTION OR IF THERE
                12382   ; WAS A CLOCK INTERRUPT.
                12383   ;
                12384   INCUFLG:  EQU H#40                      ;Inter-C.U. modifier flag
                12385   ;
019C9 FFFFF 1F210 12386   CHKOCP:  JSR     RDCUCOM               ;READ INTER-C.U. BYTE FIRST
019CA 34300 339F3 12387           RTOR    R0,R0 & BNZ INTCU2    ;If not 0, other side request
                12389
019CB 45050 F0040 12417           TSTRBIT R0,40                 ; IS THERE A DRIVE STATE CHANGE?
019CC FFFFF 331D7 12418           BOZ CK4CLK                    ; NO - CHECK FOR CLOCK INTERRUPT
019CD 0F320 E18D7 12419           CALL R0,REQSCAN               ; YES - GET THE CHANGE
019CE FFFFF 331D7 12420           BOZ CK4CLK                    ; THIS SHOULD NEVER HAPPEN
019CF 3705F F0104 12421           RDMSK@ 104                    ; ARGUMENT TO RESET STATE CHANGE -
                12422           JSR.P ISSCMD                  ;  INTERRUPT
019D0 E0401 1F624 12422 +         SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
                12423
                12424   CHKOCP1B: EQU $                         ;entry point for CDMF state_change
019D1 373F7 F0261 12425           RDRAMB R7,REQUNIT             ; UNIT # & ALL FLAG IN R7
019D2 3417F F0303 12426           ALDRAM R7,FRBUNIT             ; SAVE FOR E.C. HANDLER
019D3 37356 F0002 12427           RDMSKB R6,2                   ; SET FLAG FOR STATE CHANGE
019D4 373F5 F0262 12428           RDRAMB R5,REQSTATE            ; GET REQUESTED STATE
019D5 373F1 F0263 12429           RDRAMB R1,PRSSTATE            ;GET THE PRESENT STATE
019D6 FFFFF E1A49 12431           JMP     SCHKOCP2              ;$Do the change
019D7 45050 F0020 12455   CK4CLK:  TSTRBIT R0,20                 ; TIME OF DAY INTERRUPT?
019D8 433F6 339DD 12456           RESETR    R6 & BNZ CK4CLKB+1  ; Yes, jump; Clear flag reg.
019D9 45050 F0010 12457           TSTRBIT R0,10                 ; INTERVAL TIMER INTERRUPT?
019DA FFFFF 339DC 12458           BNZ CK4CLKB                   ; IF YES, PROCESS IT
                12460   ;
019DB E7C00 E001E 12461   CK4CLKA: CTL OCP,OCPDSEL & JMP WAIT15+1  ;$Return to Wait Loop
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
                12496   ;
019DC 37356 F0001 12497   CK4CLKB:  RDMSKB R6,1                    ; SET INTERVAL TIMER FLAG
019DD 3705F F0102 12498            RDMSKQ 102                     ; RESET CLOCK INTERRUPT FLAG
                12499              JSR.P ISSCMD
019DE E0401 1F624 12499 +          SETPGE ISSCMD_SHR_H#B & JSRP ISSCMD
019DF FFFFF E19DB 12501            JMP      CK4CLKA               ;$Return now
                12762   ;
                12763   CLKOCP6:   EQU $
019E0 433F1 1F228 12767            RESETR R1 & JSR WRTCUCOM       ;/CLEAR INTER-C.U. FLAG
019E1 FFFFF 339DB 12768            BNZ CK4CLKA                    ;/If OCP not selected, exit now
019E2 0F320 E0E15 12769            CALL     R0,SELOCP             ; SELECT OCP
019E3 FFFFF 331D7 12770            BUZ      CK4CLK                ; If selected, check for Clock
019E4 433F0 E19D7 12771            RESETR R0 & JMP CK4CLK         ; Else, return
                12772   ;
                12773   ;    OCP not available
                12774   ;
019E5 3705F F0020 12775   OCPNOSEL: RDMSKQ    20                 ; SET BIT FOR SENSE
019E6 321FF F0024 12776            QLDRAM    SEN212               ; SENSE BYTE 12 FORMAT 2
019E7 E7C00 E04AD 12777            CTL OCP,OCPDSEL & JMP READ62   ; GO SEND EQUIP CHK.
                12778
                12940   ; THIS ROUTINE HANDLES REQUESTS FOR THE WHOLE SYSTEM.  IT SETS value 83-hex IN
                12941   ;THE INTER-C.U. FLAG BYTE, THEN WAITS FOR THE BYTE TO be changed by the
                12942   ;other C.U.  If the byte is changed to 00, the routine returns (RTS).
                12943   ;If the byte is changed to another value, this request from the other C.U.
                12944   ;will be processed.
                12945   ;
019E8 3735] F0083 12946   INTCU1:   RDMSKB R1,83                   ;SET value 83
019E9 FFFFF 1F228 12947            JSR WRTCUCOM                   ; WRITE it
                12948   ;
019EA E7C00 C07FF 12949   INTCU1A:  CTL      OCP,OCPDSEL & LDCT H#7FF  ; WAIT FOR 1 MS.
019EB FFFFF FFFFF 12950            NOP
019EC FFFFF 981EB 12951            CJPCNT   EVCNTZ,$-1
019ED FFFFF 1F210 12952            JSR      RDCUCOM               ; GET FLAG WORD
019EE 65151 F0003 12953            TSTREQU R1,3                   ;/IS IT STILL 83?
019EF 333F1 331EA 12954            OR ZB,B,,R1 & BUZ INTCU1A      ;/YES, CONTINUE CHECKING
019F0 433F9 A37FF 12955            RESETR R9 & CRTS ALUZ          ;/IF IT'S 0, RETURN
019F1 45051 F0030 12956            TSTRBIT R1,30                  ;Diagnostics' data?
019F2 FFFFF 339EA 12957            BNZ INTCU1A                    ;Yes, ignore it
                12958   ;
019F3 45051 F000F 12959   INTCU2:   TSTRBIT R1,H#F                 ;/ISOLATE 4 LSB'S IN Q
019F4 0615E F19F7 12960            ADD DQ,FOUTL,MSK2,IREG & NOP INTCU3 ;/COMPUTE ACTUAL JUMP
019F5 37929 FFFFF 12962            OR       DZ,FOUTU,SEQAR,1REGH  ;$Jump according to...
019F6 34317 6F7FF 12966            RTOR R1,R7 & JIR               ;/OTHER C.U. REQUEST
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
                12967   ;
019F7 FFFFF BF1E0 12968   INTCU3:   JPOP CLKOCP6                    ;/0) IMPOSSIBLE
019F8 FFFFF E19E8 12969            JMP INTCU1                      ;/1) RE-WRITE DATA
019F9 FFFFF FFFFF 12970            NOP                             ;/2) Impossible
019FA FFFFF BF1E0 12971            JPOP CLKOCP6                    ;/3)     "     ; CHECK FOR CLOCK
019FB FFFFF FFFFF 12973            NOP                             ;$4)
019FC FFFFF FFFFF 12974            NOP                             ;$5)
019FD FFFFF FFFFF 12975            NOP                             ;$6)
019FE FFFFF FFFFF 12976            NOP                             ;$7)
019FF FFFFF FFFFF 12977            NOP                             ;$8)
01A00 FFFFF FFFFF 12978            NOP                             ;$9)
01A01 FFFFF FFFFF 12979            NOP                             ;$A)
01A02 FFFFF FFFFF 12980            NOP                             $B)
01A03 FFFFF FFFFF 12991            NOP                             ;/C) not used
01A04 FFFFF FFFFF 12992            NOP                             ;/D) not used
01A05 FFFFF FFFFF 12993            NOP                             ;/E) not used
01A06 FFFFF BF1E0 12994            JPOP CLKOCP6                    ;/F) not used
                13107   ;
                13108   ;    This routine does a Fast Select of the OCP.  If the OCP is not selected
                13109   ;immediately, the routine returns with R0 = 0.  If the OCP IS selected,
                13110   ;OCP Staus is checked for parity errors or Command Active.  If they are set,
                13111   ;the routine returns with R0 = 0.  Otherwise, the routine will read the
                13112   ;Inter-C.U. Communications OCP memory (loc. 26-hex) and, if the Valid bit
                13113   ;(80) is set, return with a value in R1 (the Valid bit is stripped off).
                13114   ;If the Valid bit is not set, the routine will return with ALU = 0.
                13115   ;
                13116   FASTCOMA: EQU $
01A07 E7000 4F002 13123   FASTCOM:  CTL OCP,OCPREQ & PSHLD 2          ;REQUEST THE OCP
01A08 C7F00 887FF 13124            RD OCP,OCPSTAT,R0 & CLPCNT EVCNTZ ;READ OCP STATUS
01A09 45150 F0001 13125            TSTBIT R0,1                     ;OCP AVAILABLE?
01A0A FFFFF 3320E 13126            BUZ FASTCOM1                    ;IF NOT, JUMP
01A0B 3710F F0267 13127            RESETRAM OCPDOWN                ;Clear OCP Down flag
01A0C 45150 F0006 13128            TSTBIT R0,6                     ;Parity error?
01A0D 37160 33214 13129            MOVE SMCST,ZERO & BUZ RDCUCOM1  ;IF NOT, GET FLAG BYTE
                13130   ;
                13131   FASTCOM1: EQU $
01A0E E7C00 FFFFF 13142   FASTCOM2: CTL OCP,OCPDSEL                 ;CANCEL OCP REQUEST...
01A0F 47360 AF7FF 13143            AND DZ,B,SMCST,R0 & RTS         ;AND RETURN (CLR. XMC PARITY)
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
                13145   ;
                13146   ;    This routine reads the OCP Inter-C.U. communications byte.
                13147   ;Registers used: R0-R4,0 & IREG.
                13148   ;
01A10 373F2 F0267 13149   RDCUCOM:  RDRANB R2,OCPDOWN                ;Is the OCP Down?
01A11 FFFFF 33A07 13150           BNZ FASTCOM                      ;Yes, fast OCP select
                13151           JSR.P OCPSELA                    ;/Select the OCP
01A12 E0404 1F032 13151 +         SETPGE OCPSELA_SHR_H#B & JSRP OCPSELA
01A13 FFFFF 33A0E 13153           BNZ     FASTCOM2                 ;$If not avail., return
                13157
01A14 3710F F00A6 13158   RDCUCOM1: RESETRAM PARCNT                  ;RESET PARITY RETRY COUNT
01A15 FFFFF 5F216 13159           LDAR $+1                         ;PARITY ERROR RETRY LOC.
01A16 B7000 F0026 13160           WRTE    OCP,OCPADR & NOP D#38    ; ADDRESS OF FLAG BYTE
01A17 C7F11 4F001 13161           RD OCP,OCPDATA,R1 & PSHLD 1
01A18 C7F11 887FF 13162           RD OCP,OCPDATA,R1 & CLPCNT EVCNTZ ; READ FLAG BYTE
                13163           JSR.P XMCTPAR                    ;CHECK CT-BUS PARITY
01A19 E0403 1F35B 13163 +         SETPGE XMCTPAR_SHR_H#B & JSRP XMCTPAR
01A1A 333F1 3321D 13164           OR ZB,B,,R1 & BOZ RDCUCON        ;IF O.K., JUMP; ANY DATA?
01A1B FFFFF 3320E 13165           BOZ FASTCOM2                     ;If not, return now
01A1C 333F1 AF7FF 13166           OR ZB,B,,R1 & RTS                ;ELSE, RETURN WITH ALU NOT = 0
                13167   ;
01A1D 45150 F0200 13168   RDCUCON:  TSTBIT R0,200                   ;Expanded OCP memory?
01A1E C7F00 33222 13169           RD OCP,OCPSTAT,R0 & BOZ RDCUCONA ;If not, jump
01A1F C7F00 FFFFF 13170           RD OCP,OCPSTAT,R0                ;Read OCP Status
01A20 45150 F0100 13171           TSTBIT R0,100                    ;OCP memory parity error?
01A21 FFFFF 33A28 13172           BNZ RDCUPAR                      ;Yes, process it
                13173   ;
01A22 34311 FFFFF 13174   RDCUCONA: RTOR R1,R1                       ;R1 = Data read
01A23 45151 F0080 13175           TSTBIT R1,80                     ;VALID BIT SET?
01A24 55341 A3880 13176           MASK DB,B,MSK1,R1 & CRTS NALUZ,H#80 ;Yes, clear it & return
01A25 45150 F0070 13177           TSTBIT R0,70                     ;State Change/TOD/Interval?
01A26 47160 A3FFF 13178           AND DZ,FOUTL,SMCST,ZERO & CRTS NALUZ ;Yes, don't de-select OCP
01A27 FFFFF E1A0E 13179           JMP FASTCOM2                     ;Else, return with OCP de-selected
                13191
                13192   RDCUPAR:  SETMBIT 40,SEN212                ;/SENSE = OCP PARITY
01A28 37350 F0040 13192 +         RDMSKB   R0,40
01A29 3410F F0024 13192 +         ALDRAM   R0,SEN212
01A2A FFFFF E1A0E 13193           JMP FASTCOM2                     ;/Return with ALU = 0
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
                13195   ;
                13196   ;    This routine writes the Inter-C.U. communications byte.
                13197   ;Upon entry:  R1 = data to be written
                13198   ;Upon exit:   ALU = 0 if able to select the OCP.
                13199   ;             ALU = 1 if unable to select the OCP.
                13200   ;Registers used:  R0,R2-R4,Q & IREG.
                13201   ;
01A2B 34311 FFFFF 13202   WRTCUCOM: RTUR R1,R1                        ;R1 = Write data
                13203             JSR.P OCPSELA                     ;/Select the OCP
01A2C E0404 1F032 13203 +           SETPGE OCPSELA_SHR_H#B & JSRP OCPSELA
01A2D 433FD 33A31 13205             RESETR   R13 & BNZ WRTCUCON       ;$If not selected, jump
01A2E B7000 F0026 13209             WRTE    OCP,OCPADR & NOP D#38     ; ADDRESS OF FLAG BYTE
01A2F 9711F 4F001 13210             WRTL    OCP,OCPDATA,R1 & PSHLD 1  ; WRITE THE DATA
01A30 9711F 887FF 13211             WRTL     OCP,OCPDATA,R1 & CLPCNT EVCNTZ
01A31 E7C00 FFFFF 13212   WRTCUCON: CTL     OCP,OCPDSEL               ; DESELECT OCP
01A32 373F0 AF267 13214             RDREGB   RAME,R0 & RTS OCPDOWN    ;$Return with OCP down status
                14067
                14068
01A33 3535F F0200 14069   SETVERFY: SETRBIT R15, H#200                ;SET NO DATA TRANSFER FLAG
                14070   ;
01A34 0F10F AF302 14071             ADD DZC,FOUTL,ZERO,RAME & RTS SEEKONL;Set the Seek Only flag & return
                14372   WAIT4SM:  EQU $
01A35 3714E F0110 14374             MOVE     MSK1,IREG & NOP H#110    ;$SM-irpt. mask to IREG
01A36 370EF FFFFF 14379             RDREGQ IREG                       ;GET SM MASK
01A37 D0FF1 FFFFF 14380             RDXFR SMC,,R1                     ;LOAD A TIMER INTO R1
01A38 45155 F2000 14381             TSTBIT R5,2000                    ;SM-WRITE ACTIVE?
                14382             BOZ.P WAITCON                     ;IF NOT, WAIT TO CONTINUE
01A39 E0400 33521 14382 +           SETPGE WAITCON_SHR_H#B & BOZP WAITCON
                14383   ;
01A3A 433F0 FFFFF 14384   WAIT4SM1: RESETR R0
01A3B 133F1 33A3D 14385             DECRB R1 & BNZ $+2                ;IF INTERRUPT, JUMP; DECR. TIMER
01A3C 46110 33A3B 14386             AND DQ,FOUTL,INTR1,ZERO & BNZ $-1 ;CHECK HSSM COMPLETE OR ERROR
01A3D C2000 33241 14387             RD HSSMA,RDINTR,R0 & BOZ $+4      ;READ INTERRUPT REG.
01A3E 45150 F0002 14388             TSTBIT R0,2                       ;SM-DONE?
01A3F E1015 33A41 14389             CTL HSSM,RSTUIRPT & BNZ $+2       ;YES, JUMP
01A40 E0082 E1A3A 14390             CTL SMC,HSSMRST & JMP WAIT4SM1    ;RESET IRPT. & WAIT FOR SM-DONE
01A41 55355 F2010 14391             RSTRBIT R5,2010                   ;RESET WRITE ACTIVE/SM-INITIATED
01A42 FFFFF E0521 14392             JMP WAITCON                       ;THEN WAIT TO CONTINUE
                14393   ;
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
01A43 373F0 F0024 14883    SETSEN12: RDRANB R0,SEN212                ;GET SENSE BYTE 12
01A44 30300 FFFFF 14884              OR AQ,B,R0,R0                   ;SET THE APPROPRIATE BIT
01A45 45350 F00FF 14885              MSKRBIT R0,H#FF
01A46 3410F AF024 14886              ALDREG R0,RAME & RTS SEN212     ;RESTORE THE SENSE BYTE & RETURN
                  15058
                  15059
01A47 FFFFF E003B 15060    SMBRDB:   JMP SMBRD
01A48 FFFFF E0044 15061    SMBWRTB:  JMP SMBWRT
```

Addr Line CLOCK AND OCP INTERRUPT PROCESSING

```
                15973   ;
01A49 65055 F0002 15974   SCHKOCP2: TSTRMSK R5,2                   ;IS REQUESTED STATE "CACHE"?
01A4A FFFFF 3324D 15975             BOZ $+3                       ;IF YES, JUMP
01A4B 65055 F0005 15976             TSTRMSK R5,5                  ;IS REQUESTED STATE "WRITE-THRU"?
01A4C FFFFF 33A4F 15977             BNZ SCHKOCP4                  ;IF NOT, CHECK FURTHER
                15978   ;
01A4D 3705F F0003 15987   SCHKOCP3: RDMSKQ 3                      ;FORCE THE STATE...
01A4E 321FF F0262 15988             QLDRAM REQSTATE               ;TO BE "BYPASS"
                15989   ;
01A4F 373F1 F0263 15990   SCHKOCP4: RDRAMB R1,PRSSTATE            ;GET THE PRESENT STATE
01A50 65051 F0001 15991             TSTRMSK R1,1                  ;IS IT 'DOWN'?
01A51 61151 33A57 15992   SCHKOCP5: XOR AB,FOUTL,R5,R1 & BNZ SCHKOCP6 ;NO, DO THE CHANGE
01A52 3437D 33257 15993             RTOR R7,R13 & BOZ SCHKOCP6    ;IF REQUESTED = DOWN, JUMP
01A53 65057 F00FF 15994             TSTRMSK R7,H#FF               ;CHANGE ALL?
01A54 34370 33257 15995             RTOR R7,R0 & BOZ SCHKOCP6     ;YES, DO IT
                15996               JSR.P SEL40                   ;SET THE BIT FOR THIS DRIVE
01A55 E0401 1F329 15996 +           SETPGE SEL40_SHR_H#B & JSRP SEL40
                15997               JSR.P SETATTN                 ;SET ATTN. PENDING THIS DRIVE
01A56 E0404 1F02B 15997 +           SETPGE SETATTN_SHR_H#B & JSRP SETATTN
                15998   ;
                15999   SCHKOCP6: EQU $
01A57 0F320 E18F6 16001             CALL    R0,SETPREZ            ;$Set the new Drive State
01A58 433F1 E19E0 16005             RESETR R1 & JMP CLKOCP6       ;NORMAL RETURN
                16006
                16087   TITLE FORMAT VERIFICATION DATA CHECKING
```

Addr Line FORMAT VERIFICATION DATA CHECKING

```
                16089   ;
01A59 45057 F0010 16090   CHKDATA:  TSTRBIT R7,10                    ;FORMAT-TO-VERIFY FLAG?
01A5A C3023 A37FF 16100           RD SCP,MODESW,R3 & CRTS ALUZ      ;IF NOT, RETURN; RD. MODE SWITCHES
01A5B 45053 F0004 16103           TSTRBIT  R3,4                     ;$Don't check data?
01A5C FFFFF A3FFF 16117           CRTS NALUZ                        ;YES, RETURN
                16118   ;
01A5D 433F6 FFFFF 16119   CHKDATA0: RESETR R6                       ;1ST WORD IS 0
01A5E 377F8 F02D9 16120           OR DZ,LSBZ,RAME,R8 & NOP WRDSTRK
01A5F 337F8 FFFFF 16121           OR ZB,LSBZ,,R8                    ;WORDS PER TRACK TO R8
                16122   ;
                16123   CHKDATA2: JSR.P ULENGTH                     ;COMPUTE & LOAD U-XFER. COUNT
01A60 E0400 1F4B4 16123 +         SETPGE ULENGTH_SHR_H#B & JSRP ULENGTH
01A61 B1F00 F0033 16124           WRTE HSSM,LDXMCJ & NOP HUNLDMD    ;START A NORMAL...
01A62 E100D FFFFF 16125           CTL HSSM,XMCJCMD                  ;U-SIDE READ
01A63 B1500 F0000 16126           WRTE HSSM,LDUARM & NOP 0          ;THE DATA STARTS...
01A64 B1400 F0000 16127           WRTE HSSM,LDUARL & NOP 0          ;AT LOC. 0000
01A65 E46F0 FFFFF 16128           CTL HIO,DENCHT                    ;DISABLE CHANNEL TAGS
01A66 E45FF FFFFF 16129           CTL HIO,CLRHSIM                   ;CLEAR ANY SIMULATE TAGS
01A67 37354 F0004 16130           RDMSKB R4,4                       ;GET THE IDR MASK
01A68 E4410 FFFFF 16131           CTL HIO,HSIMGTS                   ;ENABLE SIMULATE GATE
                16132   ;
                16136           LDWD R7,FORDAT2                   ;Get format data bits 23-0
01A69 373F7 F026B 16136 +         RDRAMB   R7,FORDAT2                ;GET 12 MSBS
01A6A D0FF7 FFFFF 16136 +         RDXFR    SMC,,R7                  ;PUT IN UPPER REGISTER
01A6B 353F7 F026C 16136 +         OR DB,B,RAME,R7 & NOP FORDAT2+1    ;OR IN 12 LSBS
01A6C 373F0 F026A 16137           RDRAMB R0,FORDAT3                 ;GET FORMAT DATA BITS 35-24
                16138   ;
                16139   CHKDATA3: JSR.P CHKTAG                      ;CHECK FOR AN IDR
01A6D E0403 1F607 16139 +         SETPGE CHKTAG_SHR_H#B & JSRP CHKTAG
01A6E 650B0 FFFFF 16140           XOR DB,0,UBR2,R0                  ;DATA BITS 35-24 GOOD?
01A6F 651A7 33A76 16141           XOR DB,FOUTL,UBR1,R7 & BNZ CHKDATA4 ;NO, JUMP; BITS 23-0 GOOD?
01A70 E457F 33A77 16142           CTL HIO,SIDACK & BNZ CHKDATA4+1   ;NO, JUMP; ACKNOWLEDGE THE IDR
01A71 133F8 FFFFF 16143           DECRB R8                          ;THIS TRACK DONE?
01A72 E45FF 33A6D 16144           CTL HIO,CLRHSIM & BNZ CHKDATA3    ;NO, CHECK THE NEXT WORD
                16145   ;
                16146           JSR.P HSWAITR                     ;HSSM WAIT LOOP
01A73 E0400 1F525 16146 +         SETPGE HSWAITR_SHR_H#B & JSRP HSWAITR
01A74 E1015 FFFFF 16147           CTL HSSM,RSTUIRPT                 ;RESET U-DONE INTERRUPT
01A75 E4420 E10CC 16148           CTL HIO,HSIMGTR & JMP ILD19       ;RETURN TO THE ILD
                16149   ;
01A76 E457F FFFFF 16150   CHKDATA4: CTL HIO,SIDACK                    ;ACKNOWLEDGE THE IDR
01A77 373B4 FFFFF 16151           RDREGB UBR2,R4                    ;GET BAD DATA BITS 35-24 INTO R4
```

Addr Line FORMAT VERIFICATION DATA CHECKING

```
01A78 373A0 FFFFF 16152            RDREGB UBR1,R0                    ;GET BAD DATA BITS 23- 0 INTO R0
01A79 E45FF FFFFF 16153            CTL HIO,CLRHSIM                   ;CLEAR SIMULATE TAGS
01A7A E4420 FFFFF 16154            CTL HIO,HSIMGTR                   ;CLEAR SIMULATE GATE
                  16155            JSR.P HSWAITR                     ;HSSM WAIT LOOP
01A7B E0400 1F525 16155 +          SETPGE HSWAITR_SHR_H#B & JSRP HSWAITR
01A7C 321FF F00B1 16167            QLDRAM MAINILD1                   ;DISPLAY BITS IN ERROR (35-24)
01A7D 34303 FFFFF 16168            RTOR R0,R3                        ;BAD DATA (BITS 23-0) TO R3
01A7E 61370 FFFFF 16169            XOR AB,B,R7,R0                    ;BITS IN ERROR (23-0) TO R0
                  16170            SVWD R0,MAINILD2                  ;Display bits in error (23-0)
01A7F 3490F F00B2 16170 +          ALDRAMU  R0,MAINILD2               ;SAVE 12 MSBS
01A80 3410F F00B3 16170 +          ALDRAM   R0,MAINILD2+1             ;SAVE 12 LSBS
                  16171    ;
                  16172            TSTMBIT LOOPFLG,MSGFLG            ;Looping-on-error flag set?
01A81 373F0 F01EF 16172 +          RDRAMB   R0,MSGFLG
01A82 45150 F0001 16172 +          TSTBIT   R0,LOOPFLG
01A83 E1015 33A8C 16173            CTL HSSM,RSTUIRPT & BNZ CHKDATA6  ;YES, JUMP; RESET U-DONE IRPT.
                  16174    ;
01A84 3414F F026D 16175    CHKDATA5: ALDRAM R4,BADDAT3                ;SAVE BAD DATA (35-24) FOR PRINTING
                  16176            SVWD R3,BADDAT2                   ;Save bad data (23-0) for printing
01A85 3493F F026E 16176 +          ALDRAMU  R3,BADDAT2                ;SAVE 12 MSBS
01A86 3413F F026F 16176 +          ALDRAM   R3,BADDAT2+1              ;SAVE 12 LSBS
01A87 0F322 E1A9A 16177            CALL R2,DATAMSG                   ;PRINT DATA ERROR MESSAGE
                  16178    ;
                  16186            TSTMBIT LOOPFLG,MSGFLG            ;Looping on error?
01A88 373F0 F01EF 16186 +          RDRAMB   R0,MSGFLG
01A89 45150 F0001 16186 +          TSTBIT   R0,LOOPFLG
                  16187            BOZ.P ILD27                       ;NO, STOP THE ILD ON ERROR
01A8A E0402 33123 16187 +          SETPGE ILD27_SHR_H#B & BOZP ILD27
01A8B FFFFF E1145 16188            JMP ILD31                         ;LOOP ON THE ERROR
                  16189    ;
01A8C 45150 F0004 16190    CHKDATA6: TSTBIT R0,MSG2                   ;Error message already printed?
01A8D FFFFF 33A91 16191            BNZ CHKDATA7                      ;YES, JUMP
01A8E 35350 F0006 16192            SETRBIT R0,MSG2+MSG1              ;SET MESSAGE SENT 1 & 2 FLAGS
01A8F 3410F F01EF 16193            ALDRAM R0,MSGFLG                  ;SAVE THEM
01A90 FFFFF E1A84 16194            JMP CHKDATA5                      ;PRINT DATA ERROR MESSAGE
                  16195    ;
01A91 651F4 F026D 16196    CHKDATA7: XOR DB,FOUTL,RAME,R4 & NOP BADDAT3 ;BAD DATA (35-24) STILL THE SAME?
01A92 FFFFF 33A84 16197            BNZ CHKDATA5                      ;NO, PRINT THE NEW BAD DATA
                  16198            LDWD R0,BADDAT2                   ;Get last bad data bits 23-0
01A93 373F0 F026E 16198 +          RDRAMB   R0,BADDAT2                ;GET 12 MSBS
01A94 D0FF0 FFFFF 16198 +          RDXFR    SMC,,R0                  ;PUT IN UPPER REGISTER
01A95 353F0 F026F 16198 +          OR DB,B,RAME,R0 & NOP BADDAT2+1    ;OR IN 12 LSBS
01A96 61130 FFFFF 16199            XOR AB,FOUTL,R3,R0                ;BAD DATA BITS 23-0 STILL THE SAME?
01A97 FFFFF 33A84 16200            BNZ CHKDATA5                      ;NO, PRINT THE NEW BAD DATA
```

Addr Line FORMAT VERIFICATION DATA CHECKING

```
01A98 373F0 F01EF 16201          RDRAMB R0,MSGFLG              ;Get message sent flags and...
01A99 FFFFF E1145 16202          JMP ILD31                     ;CONTINUE LOOPING
```

Addr Line FORMAT VERIFICATION DATA CHECKING

```
                16204
                16205
                16206   ;    THIS ROUTINE PRINTS A DATA ERROR MESSAGE.  IT FIRST PRINTS "DATA ERR:
                16207   ;GOOD =", THEN 9-HEX CHARACTERS OF REFERENCE DATA FROM MEMORY LOCATIONS
                16208   ;FORDAT3, FORDAT2 & FORDAT1.  THEN IT PRINTS ", BAD=", FOLLOWED BY 9-HEX
                16209   ;CHARACTERS OF BAD DATA FROM MEMORY LOCATIONS BADDAT3, BADDAT2 & BADDAT1.
                16210   ;
                16211   DATAMSG:  JSR.P SAVRET2                     ;SAVE THE RETURN LOC.
01A9A E0402 1F17F 16211 +         SETPGE SAVRET2_SHR_H#B & JSRP SAVRET2
01A9B 37350 F0051 16212           RDMSKB R0,D#81                    ;MSG: "DATA ERR: GOOD="
01A9C 0F322 E0E7B 16213           CALL R2,ISSCMDA1
                16214   ;
01A9D 373F0 F026A 16215           RDRAMB R0,FORDAT3                 ;GET GOOD DATA BITS 24-35 AND...
01A9E 0F322 E1E8C 16216           CALL R2,PRINT3H                   ;PRINT THEM
01A9F 373F0 F026B 16217           RDRAMB R0,FORDAT2                 ;GET GOOD DATA BITS 12-23 AND...
01AA0 0F322 E1E8C 16218           CALL R2,PRINT3H                   ;PRINT THEM
01AA1 373F0 F026C 16219           RDRAMB R0,FORDAT1                 ;GET GOOD DATA BITS 0-11 AND...
01AA2 0F322 E1E8C 16220           CALL R2,PRINT3H                   ;PRINT THEM
                16221   ;
01AA3 37350 F0052 16222           RDMSKB R0,D#82                    ;MSG: ", BAD ="
01AA4 0F322 E0E7B 16223           CALL R2,ISSCMDA1
01AA5 373F0 F026D 16224           RDRAMB R0,BADDAT3                 ;GET BAD DATA BITS 24-35 AND...
01AA6 0F322 E1E8C 16225           CALL R2,PRINT3H                   ;PRINT THEM
01AA7 373F0 F026E 16226           RDRAMB R0,BADDAT2                 ;GET BAD DATA BITS 12-23 AND...
01AA8 0F322 E1E8C 16227           CALL R2,PRINT3H                   ;PRINT THEM
01AA9 373F0 F026F 16228           RDRAMB R0,BADDAT1                 ;GET BAD DATA BITS 0-11 AND...
01AAA 0F322 E1E8C 16229           CALL R2,PRINT3H                   ;PRINT THEM
01AAB FFFFF E1185 16230           JMP RTRN2                         ;RETURN
                16231
                16232
                16233   TITLE BOOKKEEPING - PART 4
```

Addr Line BOOKKEEPING - PART 4

```
                    16235    ;
01AAC FFFFF 5F2ED   16236    SMBOOK20: LDAR SMBOOK25                    ;LOAD JUMP LOCATION
01AAD 37353 F003A   16237              RDMSKB R3,WAITOCON               ;WAIT TO CONTINUE JUMP LOC.
01AAE 370FF F02D0   16238              RDRAMQ MAXREL                    ;GET MAX. RELATIVE RECORD
01AAF 663F0 F0029   16239              XOR DQ,B,RAME,R0 & NOP RELREC    ;IS MAX.= CUR. RELATIVE REC?
01AB0 45055 F0020   16244              TSTRBIT R5,20                    ;SM-COMPLETE?
01AB1 333F0 33ADF   16245              OR ZB,B,,R0 & BNZ SMBOOK23       ;IF IT IS, JUMP
01AB2 FFFFF 332C8   16246              BOZ SMBOOK22                     ;IF MAX.= CUR. RELATIVE, JUMP
                    16247    ;
01AB3 45055 F0800   16248              TSTRBIT R5,800                   ;WRITE OPERATION?
01AB4 FFFFF 332C9   16249              BOZ SMBOOK22+1                   ;IF NOT, JUMP
                    16250    ;
01AB5 4505F F1800   16251    SMBOOK2A: TSTRBIT R15,1800                 ;ADV. CYL. OR ADV. HEAD?
01AB6 FFFFF 33ABD   16252              BNZ SMBOOK21                     ;IF EITHER, JUMP
                    16258    ;
01AB7 FFFFF 1F2FB   16259              JSR SMBOOK30                     ;ENOUGH U-DATA FOR NEXT RECORD?
01AB8 FFFFF 35ABB   16260              CJMP NALB,SMBOOK2B               ;IF THERE IS, JUMP
01AB9 FFFFF 1F317   16262              JSR SMBOOK35                     ;$CHECK FOR CDC DONE
01ABA 37341 5FD8F   16264              RDREGB MSK1,R1 & JAR WRT028      ;CHAN. DATA SLOW JUMP
                    16265    ;
01ABB 3535E F0001   16266    SMBOOK2B: SETRBIT R14,1                    ;SET WRITE NEXT FLAG
01ABC 37341 5FD93   16267              RDREGB MSK1,R1 & JAR WRT030      ;SET NEXT WRITE JUMP LOC.
                    16268    ;
01ABD 45055 F0800   16269    SMBOOK21: TSTRBIT R5,800                   ;WRITE OPERATION?
01ABE FFFFF 332C2   16270              BOZ SMBOK21B                     ;IF NOT, JUMP
01ABF FFFFF 1F2FB   16275              JSR SMBOOK30                     ;ENOUGH U-DATA FOR NEXT RECORD?
01AC0 FFFFF 352C6   16277              CJMP ALB,SMBOK21C                ;$IF NOT, JUMP
01AC1 3535E F0001   16281    SMBOK21A: SETRBIT R14,1                    ;SET WRITE NEXT FLAG
01AC2 37351 F04F4   16282    SMBOK21B: RDMSKB R1,ADVREC03               ;ADV. CYL. OR ADV. HEAD JUMP LOC.
01AC3 4505F F0800   16283              TSTRBIT R15,800                  ;ADVANCE HEAD?
01AC4 FFFFF 73AC5   16284              CJARB NALUZ,$+1                  ;NO, JUMP
01AC5 37343 5F836   16285              RDREGB MSK1,R3 & JAR WAITHD      ;ADV. HEAD WAIT-TO-CONTINUE
                    16287    ;
01AC6 FFFFF 1F317   16288    SMBOK21C: JSR SMBOOK35                     ;$CHECK FOR CDC DONE
01AC7 FFFFF E1AC2   16289              JMP SMBOK21B                     ;$IT'S NOT, CONTINUE
01AC8 3710F F0029   16291    SMBOOK22: RESETRAM RELREC                  ;RESET RELATIVE RECORD
01AC9 373F0 F02CD   16292              RDRAMB R0,RECLNTH                ;RECORD LENGTH...
                    16293              JSR.P FOUR5DN                    ;TIMES 4.5 EQUALS...
01ACA E0400 1F368   16293 +            SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
                    16294              SVWD R0,RBAU                     ;First Record Byte Address
```

Addr Line BOOKKEEPING - PART 4

```
01ACB 3490F F00AC 16294 +           ALDRAMU  R0,RBAU                ;SAVE 12 MSBS
01ACC 3410F F00AD 16294 +           ALDRAM   R0,RBAU+1              ;SAVE 12 LSBS
01ACD 55355 F0080 16295             RSTRBIT R5,80                   ;RESET U-FULL FLAG
01ACE 4505F F1000 16296             TSTRBIT R15,1000                ;ADV. CYLINDER FLAG?
01ACF FFFFF 33ABD 16297             BNZ SMBOOK21                    ;YES, JUMP
01AD0 45055 F0800 16298             TSTRBIT R5,800                  ;WRITE OPERATION?
01AD1 FFFFF 33AB5 16299             BNZ SMBOOK2A                    ;YES, JUMP
                  16300     ;
01AD2 273F0 F0206 16301             SAB DZ,B,RAME,R0 & NOP URELREC  ;GET U-SIDE RELATIVE REC.-1
01AD3 FFFFF 312D5 16302             CJMP CARRY,$+2                  ;IF A POSITIVE NO., JUMP
01AD4 273F0 F02D0 16303             SAB DZ,B,RAME,R0 & NOP MAXREL   ;ELSE, SET URELREC = MAXREL -1
                  16304     ;
01AD5 651F0 F0029 16305             XOR DB,FOUTL,RAME,R0 & NOP RELREC ;U-RELATIVE-1 = NEXT DISK RELATIVE?
01AD6 FFFFF 332E2 16306             BOZ SMBOK23A                    ;IF YES, JUMP
01AD7 133F0 FFFFF 16308             DECRB R0                        ;$DECR. U-RELATIVE -1 MORE
01AD8 FFFFF 312DA 16309             CJMP CARRY,$+2                  ;$IF POSITIVE NO., JUMP
01AD9 273F0 F02D0 16310             SAB DZ,B,RAME,R0 & NOP MAXREL   ;$ELSE, SET U-RELATIVE = MAXREL -1
01ADA 651F0 F0029 16311             XOR DB,FOUTL,RAME,R0 & NOP RELREC ;$U-RELATIVE-2 =NEXT DISK RELATIVE?
01ADB FFFFF 332E2 16312             BOZ SMBOK23A                    ;$YES, STOP READ AT END OF THIS REC.
01ADC 4505F F0800 16314             TSTRBIT R15,800                 ;ADV. HEAD FLAG?
01ADD FFFFF 33ABD 16315             BNZ SMBOOK21                    ;YES, JUMP
01ADE 37341 5FC18 16316             RDREGB MSK1,R1 & JAR READ03     ;CONTINUE READ JUMP LOC.
                  16317     ;
                  16318     ;
01ADF 45055 F0800 16319   SMBOOK23: TSTRBIT R5,800                   ;WRITE OPERATION?
01AE0 FFFFF 33AEA 16320             BNZ SMBOK23B                    ;IF YES, JUMP
01AE1 37341 5FC74 16321             RDREGB MSK1,R1 & JAR READ21     ;ELSE, STOP READ JUMP
                  16322     ;
01AE2 55355 F0010 16323   SMBOK23A: RSTRBIT R5,10                    ;RESET SM-INITIATED
01AE3 37351 F0474 16324             RDMSKB R1,READ21                ;STOP READ JUMP LOC.
01AE4 4505F F0800 16325             TSTRBIT R15,800                 ;ADV. HEAD FLAG?
01AE5 603FF 73AE6 16326             XOR AQ,B,R15,R15 & CJARB MALUZ,$+1 ;YES, RESET IT; NO, JUMP
01AE6 4505E F0200 16327             TSTRBIT R14,200                 ;DEFECTIVE TRACK?
01AE7 FFFFF 332E9 16328             BOZ $+2                         ;NO, JUMP
01AE8 37341 5FCF4 16329             RDREGB MSK1,R1 & JAR ADVREC03   ;RETURN TO ALT.+1 JUMP
01AE9 37341 5FD0D 16330             RDREGB MSK1,R1 & JAR ADVREC07   ;ADV. HEAD & STOP JUMP LOC.
                  16331     ;
01AEA 37350 F0480 16332   SMBOK23B: RDMSKB R0,READ22                ;SET WRITE COMPLETE JUMP LOC.
                  16333             JSR.P SAVIRPT                   ;SAVE IT
01AEB E0400 1F4C0 16333 +           SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
01AEC 34301 5FFFF 16334             RTOR R0,R1 & JAR
                  16356     ;
01AED 3413F F02E7 16357   SMBOOK25: ALDRAM R3,SAVXMCJ                ;SAVE NEXT OP. JUMP LOC.
01AEE 3411F F02E6 16358             ALDRAM R1,ADVLOC                ;SAVE ADV. TO NEXT RECORD JUMP
01AEF 373F2 F002B 16359   SMBOOK26: RDRAMB R2,NEXTREC               ;GET NEXT RECORD NO.
                  16360             JSR.P CHKBUSY                   ;CHECK FOR HSSM BUSY
01AF0 E0400 1F535 16360 +           SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
```

Addr Line BOOKKEEPING - PART 4

```
01AF1 91D2F FFFFF 16366            WRTL HSSM,LDDRREC,R2              ;LOAD NEXT RECORD NO.
01AF2 91F3F FFFFF 16368            WRTL HSSM,LDXMCJ,R3               ;LOAD THE NEXT JUMP LOC.
01AF3 C2000 FFFFF 16369            RD HSSMA,RDINTR,R0                ;READ THE INTERRUPT REG.
01AF4 40100 FFFFF 16370            AND AQ,FOUTL,R0,ZERO              ;CHECK AGAIN FOR BUSY
01AF5 FFFFF 33AEF 16371            BNZ SMBOOK26                      ;IF YES, RELOAD THE STUFF
01AF6 FFFFF AF7FF 16372            RTS                               ;ELSE, RETURN
                  16373    ;
                  16374    SMBOOK27: JSR.P DLAY7US                   ;WAIT FOR SM-SIDE TO START
01AF7 E0400 1F52E 16374 +          SETPGE DLAY7US_SHR_H#B & JSRP DLAY7US
01AF8 373F3 F02E7 16375            RDRAMB R3,SAVXMCJ                 ;GET THE NEXT OP. JUMP LOC
                  16382    ;
01AF9 FFFFF 1F2EF 16383    SMBOOK28: JSR SMBOOK26                    ;LOAD THE NEXT RECORD NO.
01AFA FFFFF E04C8 16384            JMP RDSMADDR                      ;LOAD NEXT SM-ADDRESS
                  16385
                  16386
01AFB 3710F F00A6 16387    SMBOOK30: RESETRAM PARCNT                 ;RESET PARITY RETRY COUNT
                  16388            LDWD R2,RBAU                      ;Next Record Byte Addr. to R2
01AFC 373F2 F00AC 16388 +          RDRAMB   R2,RBAU                 ;GET 12 MSBS
01AFD D0FF2 FFFFF 16388 +          RDXFR    SMC,,R2                  ;PUT IN UPPER REGISTER
01AFE 353F2 F00AD 16388 +          OR DB,B,RAME,R2 & NOP RBAU+1     ;OR IN 12 LSBS
                  16389    ;
                  16390    SMBOOK31: JSR.P CHKBUSY                   ;CHECK FOR INTERFACE BUSY
01AFF E0400 1F535 16390 +          SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
01B00 C2000 FFFFF 16391            RD HSSMA,RDINTR,R0
01B01 45150 F0002 16392            TSTBIT R0,2                       ;SM-DONE INTERRUPT
                  16393            CJPOP.P NALUZ,SMIRPT              ;IF YES, PROCESS IT
01B02 E0400 B3B7C 16393 +          SETPGE SMIRPT_SHR_H#B & CJPOPP NALUZ,SMIRPT
01B03 37160 4F001 16394            MOVE SMCST,ZERO & PSHLD 1         ;RESET XMC PARITY ERRORS
01B04 C1030 887FF 16395            RD HSSM,RDUARM,R0 & CLPCNT EVCNTZ
01B05 C1030 4F001 16396            RD HSSM,RDUARM,R0 & PSHLD 1       ;CUR. U-ADDR. MSB'S TO R0
01B06 C1021 887FF 16397            RD HSSM,RDUARL,R1 & CLPCNT EVCNTZ ;CUR. U-ADDR. LSB'S TO R1
01B07 C2004 FFFFF 16398            RD HSSMA,RDINTR,R4                ;READ INTERRUPTS
01B08 40140 C0003 16399            AND AQ,FOUTL,R4,ZERO & LDCT 3     ;WAS BUSY SET?
01B09 FFFFF 33AFF 16400            BNZ SMBOOK31                      ;IF YES, RE-READ THE ADDRESS
                  16401    ;
01B0A 37364 FFFFF 16402            RDREGB SMCST,R4                   ;READ XMC STATUS
01B0B 45054 F0004 16403            TSTRBIT R4,4                      ;CT-BUS PARITY?
01B0C FFFFF 33B12 16404            BNZ SMBOOK32                      ;YES, PROCESS IT
                  16405    ;
01B0D 01700 9B30D 16406            ADD AB,LSBZ,R0,R0 & CJPCNT EVCNTZ,$ ;MOVE THE MSB'S INTO POSITION
01B0E 31310 FFFFF 16407            OR AB,B,R1,R0                     ;COMBINE LSB'S & MSB'S IN R0
01B0F 45055 F0080 16408            TSTRBIT R5,B0                     ;U-FULL FLAG SET?
01B10 29102 A37FF 16409            SUBB AB,FOUTL,R0,R2 & CRYS ALUZ   ;U-ADDR.= OR > RECORD BYTE ADDR?
01B11 29120 AF7FF 16410            SUBB AB,FOUTL,R2,R0 & RTS         ;RECORD BYTE ADDR.> U-ADDR?
                  16411    ;
                  16412    ;    CT-BUS PARITY ERROR READING U-ADDRESS FROM HSSM.
                  16413    ;
```

Addr Line BOOKKEEPING - PART 4

```
01B12 0F0FF F00A6 16414    SMBOOK32: ADD DZC,Q,RAME & NOP PARCNT        ;GET PARITY RETRY COUNT +1
01B13 321FF F00A6 16415              QLDRAM PARCNT                      ;SAVE IT
01B14 2E140 F000A 16416              SUBB DQ,FOUTL,MSK1,ZERO & NOP D#10 ;RETRY COUNT < 10?
01B15 FFFFF 35AFF 16417              CJNP NALB,SMBOOK31                 ;YES, TRY AGAIN
01B16 34343 BF362 16418              RTOR R4,R3 & JPOP HSSMPAR          ;ELSE, EQUIP. CHECK
                  16420
                  16421    ;$  IF AN SM-WRITE IS NOT ACTIVE AND TRUNCATION HAS OCCURRED, NORMAL ENDING
                  16422    ;$WILL HAPPEN NOW.  OTHERWISE, THIS ROUTINE WILL RETURN.
                  16423    ;
01B17 45055 F2000 16424    SMBOOK35: TSTRBIT R5,2000                    ;$SM-WRITE ACTIVE?
01B18 4504F A3802 16425              AND DB,Q,MSK1,R15 & CRTS NALUZ,H#2 ;$YES, RETURN; TRUNCATION?
01B19 FFFFF A37FF 16426              CRTS ALUZ                          ;$NO, RETURN
01B1A FFFFF 8F7FF 16427              POP                                ;$ELSE, POP YOUR STACK...
                  16428              JPOP.P READ22                      ;$TWICE & TAKE NORMAL EXIT
01B1B E0400 BF480 16428 +            SETPGE READ22_SHR_H#B & JPOPP READ22
                  16430
                  16431
                  16432    TITLE VERY MISCELLANEOUS ROUTINES
```

```
 Addr             Line VERY MISCELLANEOUS ROUTINES


                  16434
                  16435
01B1C FFFFF 33B1E 16436    WRT035A:  BNZ $+2                          ;IF START WORD NOT 0, JUMP
01B1D 35355 F0080 16437              SETRBIT R5,80                    ;SET U-FULL FLAG
                  16438    ;
01B1E 373F2 F02DB 16439              RDRAMB R2,RECSREAD               ;NO. OF RECORDS WRITTEN
01B1F E1005 33B22 16440              CTL HSSM,WRTCONT & BNZ $+3       ;IF ANY, JUMP
01B20 35355 F0008 16441              SETRBIT R5,8                     ;ELSE, SET U-SIDE WAITING FLAG...
01B21 FFFFF E057D 16442              JMP WRT016                       ;AND CHECK FOR SM-DONE
                  16443    ;
01B22 373F1 F02CD 16444              RDRAMB R1,RECLNTH                ;RECS. WRITTEN X RECORD LENGTH...
01B23 34353 1F346 16445              RTOR R5,R3 & JSR MULTPLYC        ; = WORDS WRITTEN (R0)
01B24 34335 FFFFF 16446              RTOR R3,R5                       ;RESTORE FLAG REG.
01B25 3710F F02DB 16447              RESETRAM RECSREAD                ;RESET RECORDS WRITTEN
01B26 377F1 F02CF 16448              OR DZ,LSBZ,RAME,R1 & NOP BUFCAP
01B27 337F1 FFFFF 16449              OR ZB,LSBZ,,R1                   ;BUFFER CAPACITY TO R1
01B28 291A0 FFFFF 16450              SUBB AB,FOUTL,R10,R0             ;IF WORDS WRITTEN ARE > U-XFER...
01B29 01006 3532D 16451              ADD AB,Q,R0,R6 & CJMP ALB,WRT035B ;JUMP; WRITTEN + START WORD = Q
                  16452    ;
01B2A 18318 FFFFF 16453              SUBA AQ,B,R1,R8                  ;Q - BUFFER CAPACITY = UXFER
01B2B FFFFF 3432D 16454              CJMP AGB,WRT035B                 ;IF Q > BUFFER CAPACITY, JUMP
01B2C 34308 E1B32 16455              RTOR R0,R8 & JMP WRT035D         ;ELSE, SET UXFER = WORDS WRITTEN
                  16456    ;
01B2D 010A6 FFFFF 16457    WRT035B:  ADD AB,Q,R10,R6                  ;START WORD + U-COUNT = Q
01B2E 28110 FFFFF 16458              SUBB AQ,FOUTL,R1,ZERO            ;IF Q < BUFFER CAPACITY...
01B2F 29016 35B31 16459              SUBB AB,Q,R1,R6 & CJMP NALB,$+2  ;JUMP; BUF. CAP.- START WORD = Q
01B30 323F8 E054F 16461              QTOR R8 & JMP WRT005+1           ;$XFER. DATA TO END OF BUFFER
01B31 343A8 FFFFF 16462              RTOR     R10,R8                  ;$Remaining U-xfer. to R8
                  16483    ;
01B32 370FF F0029 16484    WRT035D:  RDRAMQ RELREC                    ;GET NEXT RELATIVE RECORD NO.
01B33 333F6 33B37 16485              OR ZB,B,,R6 & BNZ $+4            ;IF NON-ZERO, JUMP
                  16486              BNZ.P WRT005+1                   ;IF U-WORD NO. NON-ZERO, JUMP
01B34 E0400 33D4F 16486 +            SETPGE WRT005+1_SHR_H#B & BNZP WRT005+1
01B35 5535E F0001 16487              RSTRBIT R14,1                    ;RESET WRITE NEXT FLAG
01B36 55355 F0080 16488              RSTRBIT R5,80                    ;RESET U-FULL FLAG
01B37 FFFFF E054F 16489              JMP WRT005+1                     ;START THE U-SIDE AGAIN
                  16490
                  16491
01B38 5535E F0800 16492    ILD1RTZ:  RSTRBIT R14,800                  ;RESET SECOND DRIVE FLAG
                  16493              JSR.P GETFLAG                    ;GET FLAGS FOR THIS DRIVE
01B39 E0402 1F18B 16493 +            SETPGE GETFLAG_SHR_H#B & JSRP GETFLAG
01B3A 45058 F0400 16494              TSTRBIT R8,400                   ;TWO DRIVE UNIT?
01B3B FFFFF 33344 16495              BOZ ILD1RTZA                     ;IF NOT, RTZ JUST THIS DRIVE
                  16496    ;
01B3C 0F322 E1239 16497              CALL R2,ILDRUN                   ;SELECT THE LOWER DRIVE
01B3D B2200 F0040 16498              WRTE HSSMA,LDDRBO & NOP H#40     ;BUS OUT TO RE-ZERO
01B3E FFFFF 4F006 16499              PSHLD 6
01B3F E100A 887FF 16500              CTL HSSM,SETAG3 & CLPCNT EVCNTZ  ;RE-ZERO THE DRIVE
01B40 E1016 FFFFF 16501              CTL HSSM,RSTAG3
```

```
 Addr             Line VERY MISCELLANEOUS ROUTINES

                  16502             JSR.P RELDRIV                     ;RELEASE & DE-SELECT THE DRIVE
01B41 E0401 1F60F 16502 +           SETPGE RELDRIV_SHR_H#B & JSRP RELDRIV
                  16503    ;
01B42 3535E F0800 16504             SETRBIT R14,800                   ;SET SECOND DRIVE FLAG
01B43 373FB F00B8 16505             RDRAMB R11,STCYL                  ;GET START CYL. FOR POLLING LATER
                  16506    ;
01B44 0F322 E115A 16507    ILD1RTZA: CALL R2,ILDSK0                    ;RE-ZERO THE DRIVE
01B45 FFFFF E136E 16508             JMP ILD120A                       ;RETURN TO FORMAT
                  16509
                  16510
01B46 FFFFF E02C0 16511    MULTPLYC: JMP MULTPLY
```

Addr Line VERY MISCELLANEOUS ROUTINES

```
                16513
01B47 37353 F0014 16514    SENS80:  RDMSKB R3,SEN411                ;NEXT BYTE TO ASSEMBLE
01B48 0F3F0 F02D3 16515             ADD DZC,B,RAME,R0 & NOP MAXCYLA  ;NO. OF CYLINDERS TO R0
01B49 45050 F00FF 16516             TSTRBIT R0,H#FF                  ;ISOLATE 8 LSB'S IN Q...
01B4A 60500 4F006 16517             XOR AQ,RSBZ,R0,R0 & PSHLD 6      ;AND RIGHT SHIFT...
01B4B 335F0 887FF 16518             RSHFTZ R0                        ;MSB'S INTO POSITION
                16519               JSR.P SENS25                     ;SAVE BYTE 10
01B4C E0401 1F4D9 16519 +           SETPGE SENS25_SHR_H#B & JSRP SENS25
01B4D 323F0 FFFFF 16520             QTOR R0                          ;LSB'S BACK TO R0
                16521               JSR.P SENS25                     ;SAVE BYTE 11
01B4E E0401 1F4D9 16521 +           SETPGE SENS25_SHR_H#B & JSRP SENS25
                16522    ;
01B4F 0F3F0 F02D5 16523             ADD DZC,B,RAME,R0 & NOP MAXHDA   ;NO. OF HEADS TO R0
                16524               JSR.P SENS25                     ;SAVE BYTE 12
01B50 E0401 1F4D9 16524 +           SETPGE SENS25_SHR_H#B & JSRP SENS25
                16525    ;
01B51 373F0 F02D6 16526             RDRAMB R0,AMPREC                 ;NO. OF RECORDS TO R0
                16527               JSR.P SENS25                     ;SAVE BYTE 13
01B52 E0401 1F4D9 16527 +           SETPGE SENS25_SHR_H#B & JSRP SENS25
01B53 45058 F000E 16531             TSTRBIT R8,H#E                   ;ISOLATE DRIVE TYPE
01B54 6605F F0008 16532             TSTQMSK 8                        ;DRIVE TYPE 4 (SWALLOW-5)?
                16533               BNZ.P SENS18A                    ;NO, CONTINUE
01B55 E0401 33CC3 16533 +           SETPGE SENS18A_SHR_H#B & BNZP SENS18A
01B56 370FF F02D3 16534             RDRAMQ MAXCYLA                   ;GET MAX. USEABLE CYLINDER
01B57 66150 F02DA 16535             XOR DQ,FOUTL,MSK2,ZERO & NOP D#730 ;Is it 730?
                16536               BNZ.P SENS18A                    ;IF NOT, CONTINUE
01B58 E0401 33CC3 16536 +           SETPGE SENS18A_SHR_H#B & BNZP SENS18A
01B59 37350 F0034 16537             RDMSKB R0,34                     ;REV. K EMULATION = 8434
01B5A FFFFF E0CC4 16538             JMP SENS19
```

Addr Line VERY MISCELLANEOUS ROUTINES

```
                16541
                16542
01B5B 3706F FFFFF 16543    XMCTPAR:  RDREGQ SMCST                    ;READ XMC STATUS
01B5C 4605F F0004 16544              TSTQBIT 4                       ;CT-BUS PARITY ERROR?
01B5D 320FF A37FF 16545              OR ZQ,Q & CRTS ALUZ             ;NO, RETURN
                16546    ;
01B5E 370FF F00A6 16547    XMCTPAR1: RDRAMQ   PARCNT                 ;Get the retry count
01B5F 320FF A3FFF 16548              OR       ZQ,Q & CRTS NALUZ      ;If not 0, return with ALU not = 0
01B60 0A1FF 8F0A6 16549              ADD ZQC,FOUTL,,RAME & POP PARCNT ;Incr. retry count and...
01B61 FFFFF 5FFFF 16550              JAR                             ;JUMP TO RETRY
                16551    ;
01B62 FFFFF 1F235 16552    HSSMPAR:  JSR WAIT4SM                     ;CHECK FOR SM-DONE
01B63 3705F F0004 16553              RDMSKQ 4                        ;CT-HSSM PARITY
01B64 321FF F0023 16554              QLDRAM SEN211                   ;SAVE ERROR STATUS FOR SENSE
01B65 37353 F0004 16555              RDMSKB R3,4                     ;SET PARITY ERROR STATUS
01B66 FFFFF E03BA 16556              JMP IRPT12+1                    ;SEND UNIT CHECK
                16557    ;
                16558    ;    THIS ROUTINE CLEARS THE SAVED SECTOR NUMBER FOR A DRIVE.  ENTRY IS TO
                16559    ;'CLRSECT' IF R13 = LOGICAL DRIVE NO., OR TO 'CLRSECT1' IF R13 = PHYSICAL
                16560    ;DRIVE NUMBER..
                16561    ;
01B67 340DF E1B69 16562    CLRSECT:  RTOQ R13 & JMP $+2              ;LOGICAL DRIVE TO Q
                16563    CLRSECT1: JSR.P LOGICAL                   ;GET LOGICAL DRIVE INTO Q
01B68 E0400 1F1F9 16563 +            SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
01B69 4605F F003F 16564              TSTQBIT 3F                      ;ISOLATE DRIVE NO. ONLY
01B6A 0615E F03C3 16565              ADD DQ,FOUTL,MSK2,IREG & NOP TRKNBR ;LOC. OF SAVED SECTOR THIS DRIVE
01B6B 3710D FFFFF 16566              MOVE ZERO,RAMI                  ;CLEAR IT
01B6C 0615E F0474 16567              ADD DQ,FOUTL,MSK2,IREG & NOP LASTSECT ;Loc. of Last sector the drive
01B6D 3710D AF7FF 16568              MOVE     ZERO,RAMI & RTS        ;Clear it & return
```

```
 Addr             Line VERY MISCELLANEOUS ROUTINES


                  16591
01B6E 373F2 F0245 16592    ILD109:   RDRAMB R2,MAXCYLDI              ;GET MAX. DRIVE CYLINDER
01B6F 373F1 F0088 16593              RDRAMB R1,STCYL                 ;GET STARTING CYLINDER
                  16594              BNZ.P BADCYL                    ;IF IT'S NOT 0, BAD ENTRY
01B70 E0402 339F6 16594 +            SETPGE BADCYL_SHR_H#B & BNZP BADCYL
01B71 61120 FFFFF 16595              XOR AB,FOUTL,R2,R0              ;END CYL.= MAX. DRIVE CYL?
                  16596              BNZ.P BADCYL                    ;IF NOT, BAD ENTRY
01B72 E0402 339F6 16596 +            SETPGE BADCYL_SHR_H#B & BNZP BADCYL
                  16597    ;
                  16598    ;    IF WE FALL THRU HERE WE ARE DOING A FULL PACK FORMAT, INCLUDING THE
                  16599    ;ALTERNATE CYLINDERS
                  16600    ;
01B73 3710F F00BB 16601              RESETRAM STHD                   ;STARTING HEAD IS 0
01B74 373F1 F020D 16602              RDRAMB R1,MAXHDI                ;GET MAX. HEAD NO.
01B75 3411F F00BC 16603              ALDRAM R1,NDHD                  ;SAVE MAX. HEAD
01B76 FFFFF E1353 16604              JMP ILD115                      ;CHECK RECORD LENGTH & EMULATION
                  16605
                  16682    TITLE CONFIDENCE DIAGNOSTICS LINKAGE
```

```
 Addr            Line CONFIDENCE DIAGNOSTICS LINKAGE


                 16726
                 16727    ;  THIS ROUTINE IS THE RETURN LINKAGE FROM THE CONFIDENCE DIAGNOSTICS.
                 16728    ;
01B77 0F320 E0E15 16729    DIAGRTN:  CALL R0,SELOCP                    ;SELECT THE OCP
01B78 C3017 FFFFF 16731              RD       SCP,PRGTGL,R7           ;$Read the Toggle reg.
01B79 E7040 FFFFF 16732              CTL      OCP,RCMDACT             ;$Reset OCP Command active
01B7A 3710F F00BA 16784              RESETRAM ILDFLG                  ;Reset ILD flags
01B7B E7C00 E02F4 16785              CTL OCP,OCPDSEL & JMP CLR02C     ;CONTINUE with Clear routine
                 16786
                 17154    TITLE ANGULAR POSITIONING ROUTINES (R.P.S.)
```

Addr Line ANGULAR POSITIONING ROUTINES (R.P.S.)

```
                17156
                17157   ;VALSECT:  EQU H#400                        ;VALID SECTOR NUMBER
                17158   ;SECTIME: EQU H#4B4 thru H#4D3  Display 591 - 598  Sector timers
                17159   ;                                               (1 loc. per drive)
                17160   ;
01B7C FFFFF E003B 17161   SMBRDD:   JMP SMBRD
01B7D FFFFF E0044 17162   SMBWRTD:  JMP SMBWRT
                17163   ;
                17164   SECSAV:   JSR.P GETDRBIT                    ;GET THE BIT FOR THIS DRIVE
01B7E E0401 1F348 17164 +         SETPGE GETDRBIT_SHR_H#B & JSRP GETDRBIT
01B7F 373F1 F00BA 17178           RDRAMB R1,ILDFLG                  ;GET ILD FLAGS
01B80 45051 F0004 17179           TSTRBIT R1,4                      ;ILD RUNNING?
01B81 4504F A3B80 17183           AND DB,Q,MSK1,R15 & CRTS NALUZ,H#80 ;YES, RETURN; SEEK ONLY FLAG?
01B82 370FF A3271 17184           RDREGQ RAME & CRTS ALUZ,EFSTACK+1 ;NO, RETURN; GET LAST FUNCT. WORD
01B83 4605F F0FC0 17185           TSTQBIT H#FC0                     ;ISOLATE THE COMMAND CODE
01B84 6605F F0AC0 17186           TSTQMSK H#AC0                     ;"SECTOR POSITION RELATIVE"?
01B85 FFFFF A3FFF 17187           CRTS NALUZ                        ;NO, RETURN
                17188   ;
                17195           JSR.P    GETFLAG                  ;Get flags for this drive into R8
01B86 E0402 1F188 17195 +         SETPGE GETFLAG_SHR_H#B & JSRP GETFLAG
01B87 45048 A300E 17196           AND DB,Q,MSK1,R8 & CRTS ALUZ,H#E  ;NONE, RETURN; DRIVE TYPE to Q
01B88 FFFFF 5F3C4 17197           LDAR     SECSAV3                  ;Jump loc. to AR
01B89 45358 F0030 17198           MSKRBIT R8,30                     ;ISOLATE REC. LENGTH IN R8
01B8A 66150 F000C 17199           TSTQEQU H#C                       ;CDC-368 DRIVE (TYPE 6)?
01B8B FFFFF 33BE1 17204           BNZ SECSAV8                       ;IF NOT, CHECK FOR FUJITSU
                17205   ;
01B8C 65058 F0010 17207           TSTRMSK R8,10                     ;$RECORD LENGTH = 116?
01B8D 3409F A3FFF 17208           RTOQ R9 & CRTS NALUZ              ;$IF NOT, ERROR, RETURN; REC. TO Q
01B8E 433F1 73B8F 17213           RESETR R1 & CJARB NALUZ,$+1       ;IF RECORD 0, JUMP TO SECSAV3
                17214   ;
01B8F 0615E F1B91 17216           ADD DQ,FOUTL,MSK2,IREG & NOP $+D#2 ;$COMPUTE THE 116 JUMP...
01B90 37929 FFFFF 17220           OR DZ,FOUTU,SEQAR,IREGH           ;LOCATION ACCORDING TO RECORD NO.
01B91 FFFFF 6F7FF 17221           JIR
                17222   ;                                           ;RECORD
01B92 37341 5F800 17223           RDREGB MSK1,R1 & JAR D#0          ;   1
01B93 37341 5F801 17224           RDREGB MSK1,R1 & JAR D#1          ;   2
01B94 37341 5F802 17225           RDREGB MSK1,R1 & JAR D#2          ;   3
01B95 37341 5F804 17226           RDREGB MSK1,R1 & JAR D#4          ;   4
```

Addr Line ANGULAR POSITIONING ROUTINES (R.P.S.)

```
01B96 37341 5F805 17228            RDREGB MSK1,R1 & JAR D#5          ;$  5
01B97 37341 5F806 17229            RDREGB MSK1,R1 & JAR D#6          ;$  6
01B98 37341 5F807 17230            RDREGB MSK1,R1 & JAR D#7          ;$  7
01B99 37341 5F808 17231            RDREGB MSK1,R1 & JAR D#8          ;$  8
01B9A 37341 5F80A 17232            RDREGB MSK1,R1 & JAR D#10         ;$  9
01B9B 37341 5F80B 17233            RDREGB MSK1,R1 & JAR D#11         ;$ 10
01B9C 37341 5F80C 17234            RDREGB MSK1,R1 & JAR D#12         ;$ 11
01B9D 37341 5F80D 17235            RDREGB MSK1,R1 & JAR D#13         ;$ 12
01B9E 37341 5F80F 17236            RDREGB MSK1,R1 & JAR D#15         ;$ 13
01B9F 37341 5F810 17237            RDREGB MSK1,R1 & JAR D#16         ;$ 14
01BA0 37341 5F811 17238            RDREGB MSK1,R1 & JAR D#17         ;$ 15
01BA1 37341 5F812 17239            RDREGB MSK1,R1 & JAR D#18         ;$ 16
01BA2 37341 5F814 17240            RDREGB MSK1,R1 & JAR D#20         ;$ 17
01BA3 37341 5F815 17241            RDREGB MSK1,R1 & JAR D#21         ;$ 18
01BA4 37341 5F816 17242            RDREGB MSK1,R1 & JAR D#22         ;$ 19
01BA5 37341 5F817 17243            RDREGB MSK1,R1 & JAR D#23         ;$ 20
01BA6 37341 5F819 17244            RDREGB MSK1,R1 & JAR D#25         ;$ 21
01BA7 37341 5F81A 17245            RDREGB MSK1,R1 & JAR D#26         ;$ 22
01BA8 37341 5F81B 17246            RDREGB MSK1,R1 & JAR D#27         ;$ 23
01BA9 37341 5F81C 17247            RDREGB MSK1,R1 & JAR D#28         ;$ 24
01BAA 37341 5F81E 17248            RDREGB MSK1,R1 & JAR D#30         ;$ 25
01BAB 37341 5F81F 17249            RDREGB MSK1,R1 & JAR D#31         ;$ 26
01BAC 37341 5F820 17250            RDREGB MSK1,R1 & JAR D#32         ;$ 27
01BAD 37341 5F821 17251            RDREGB MSK1,R1 & JAR D#33         ;$ 28
01BAE 37341 5F823 17252            RDREGB MSK1,R1 & JAR D#35         ;$ 29
01BAF 37341 5F824 17253            RDREGB MSK1,R1 & JAR D#36         ;$ 30
01BB0 37341 5F825 17254            RDREGB MSK1,R1 & JAR D#37         ;$ 31
01BB1 37341 5F826 17255            RDREGB MSK1,R1 & JAR D#38         ;$ 32
01BB2 37341 5F828 17256            RDREGB MSK1,R1 & JAR D#40         ;$ 33
01BB3 37341 5F829 17257            RDREGB MSK1,R1 & JAR D#41         ;$ 34
01BB4 37341 5F82A 17258            RDREGB MSK1,R1 & JAR D#42         ;$ 35
01BB5 37341 5F82B 17259            RDREGB MSK1,R1 & JAR D#43         ;$ 36
01BB6 37341 5F82D 17260            RDREGB MSK1,R1 & JAR D#45         ;$ 37
01BB7 37341 5F82E 17261            RDREGB MSK1,R1 & JAR D#46         ;$ 38
01BB8 37341 5F82F 17262            RDREGB MSK1,R1 & JAR D#47         ;$ 39
01BB9 37341 5F830 17263            RDREGB MSK1,R1 & JAR D#48         ;$ 40
01BBA 37341 5F832 17264            RDREGB MSK1,R1 & JAR D#50         ;$ 41
01BBB 37341 5F833 17265            RDREGB MSK1,R1 & JAR D#51         ;$ 42
01BBC 37341 5F834 17266            RDREGB MSK1,R1 & JAR D#52         ;$ 43
01BBD 37341 5F835 17267            RDREGB MSK1,R1 & JAR D#53         ;$ 44
01BBE 37341 5F837 17268            RDREGB MSK1,R1 & JAR D#55         ;$ 45
01BBF 37341 5F838 17269            RDREGB MSK1,R1 & JAR D#56         ;$ 46
01BC0 37341 5F839 17270            RDREGB MSK1,R1 & JAR D#57         ;$ 47
01BC1 37341 5F83A 17271            RDREGB MSK1,R1 & JAR D#58         ;$ 48
01BC2 37341 5F83C 17272            RDREGB MSK1,R1 & JAR D#60         ;$ 49
01BC3 37341 5F83D 17273            RDREGB MSK1,R1 & JAR D#61         ;$ 50
                    17346   ;
01BC4 34315 5F3CE 17348   SECSAV3: RTOR R1,R5 & LDAR SECSAV4A        ;$POSITIVE SECTOR JUMP LOC.
01BC5 45357 F003F 17352            MSKRBIT R7,3F                     ;ISOLATE BACKING FACTOR
```

Addr Line ANGULAR POSITIONING ROUTINES (R.P.S.)

```
01BC6 333F6 333DF 17353            OR ZB,B,,R6 & BOZ SECSAV6          ;IF 0, WE WILL ASSUME THE...
                  17354    ;                                            FUNCTION IS A WRITE
                  17355    ;
01BC7 370FF F03EF 17356            RDRAMQ BAKFACTR                    ;GET SAVED BACKING FACTOR
01BC8 28170 333CA 17357            SUBB AQ,FOUTL,R7,ZERO & BOZ $+2    ;IS IT < CURRENT BACKING FACTOR?
01BC9 333F6 343DF 17358            OR ZB,B,,R6 & CJMP AGB,SECSAV6     ;YES, NEXT FUNCTION IS A WRITE
01BCA 3417F F03EF 17359            ALDRAM R7,BAKFACTR                 ;SAVE MIN. BACKING FACTOR (READ)
                  17360    ;
                  17361    ;   NEXT FUNCTION ON THIS DRIVE WILL BE A 'READ'.
                  17362    ;
01BCB 1D345 F0006 17367            SUBA DB,B,MSK1,R5 & NOP D#6        ;DECR. SECTOR NO. BY 6
01BCC FFFFF 71BCD 17368            CJARB NCARRY,SECSAV4               ;IF A POSITIVE NO., JUMP
01BCD 05345 F0040 17369    SECSAV4:  ADD DB,B,MSK1,R5 & NOP D#64        ;MAKE THE SECTOR POSITIVE
                  17370    ;
01BCE 35355 F0400 17371    SECSAV4A: SETRBIT R5,VALSECT                ;INCLUDE THE VALID BIT
01BCF 6505A F0005 17372            TSTRMSK R10,5                      ;IS THE DRIVE ON CYLINDER?
01BD0 433F1 33BD4 17373            RESETR   R1 & BNZ SECSAV5          ;No, jump;  Reset current sector
01BD1 FFFFF 1F485 17374            JSR RDSECT                         ;READ CURRENT SECTOR
01BD2 61151 FFFFF 17375            XOR AB,FOUTL,R5,R1                 ;CURRENT SECTOR = DESIRED SECTOR?
                  17377            CJPUP.P ALUZ,SEEK10                ;$YES, SEND CHAN. END/DEV. END
01BD3 E0400 B3183 17377 +          SETPGE SEEK10_SHR_H#B & CJPUPP ALUZ,SEEK10
                  17381    ;
01BD4 3431A FFFFF 17382    SECSAV5:  RTOR     R1,R10                  ;Current sector to R10
01BD5 37352 F03FC 17383            RDMSKB R2,RPSPEND                  ;LOC. OF R.P.S. PENDING
                  17384            JSR.P    ADDABIT                   ;Add this drive to R.P.S. pending
01BD6 E0400 1F1FF 17384 +          SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
                  17385            JSR.P LOGICAL                      ;LOGICAL DRIVE NO. TO Q
01BD7 E0400 1F1F9 17385 +          SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
01BD8 0615E BF3C3 17386            ADD DQ,FOUTL,MSK2,IREG & POP TRKNBR ;LOC. OF SECTOR SAVE THIS DRIVE
01BD9 3415D FFFFF 17387            ALDREG R5,RAMI                     ;SAVE DESIRED SECTOR
                  17388    ;
01BDA 0615E F0474 17394            ADD DQ,FOUTL,MSK2,IREG & NOP LASTSECT ;Loc. of Last sector this drive
01BDB 341A0 FFFFF 17395            ALDREG   R10,RAMI                  ;Save Last sector
                  17396    ;
01BDC 0615E F04B4 17397            ADD DQ,FOUTL,MSK2,IREG & NOP SECTIME ;LOC. OF TIMER FOR THIS DRIVE
01BDD 37350 F0100 17398            RDMSKB   R0,D#256                  ;Set timer to 256
01BDE 3410D E016C 17399            ALDREG R0,RAMI & JMP SEEK06C       ;SAVE TIMER & SEND CHANNEL END
                  17400    ;
                  17401    ;    NEXT FUNCTION ON THIS DRIVE WILL BE A 'WRITE'.
                  17402    ;
01BDF 1D345 F0007 17404    SECSAV6:  SUBA DB,B,MSK1,R5 & NOP D#7        ;$DECR. SECTOR BY 7
01BE0 FFFFF 71BCD 17411            CJARB NCARRY,SECSAV4               ;IF A POSITIVE NO., SECSAV4A
```

Addr Line ANGULAR POSITIONING ROUTINES (R.P.S.)

```
                17420
01BE1 FFFFF 5F3CE 17421   SECSAV8:  LDAR SECSAV4A
01BE2 66150 F0006 17422            TSTQEQU  6                    ;FUJI M2333K DRIVE (TYPE 3)?
01BE3 FFFFF 333E6 17423            BOZ $+3                       ;YES, JUMP
01BE4 66150 F000A 17424            TSTQEQU  H#A                  ;FUJI SWALLOW-4 DRIVE (TYPE 5)?
01BE5 FFFFF 33BF2 17425            BNZ SECSAV20                  ;NO, CHECK FOR SWALLOW-5
                17427   SECSAV8A:   EQU   $
01BE6 45357 F003F 17431            MSKRBIT  R7,3F                ;ANY BACKING FACTOR?
01BE7 14391 333F0 17432            SBA AZ,8,R9,R1 & BOZ SECSAV10 ;NO, ASSUME NEXT FUNCT.=WRITE
01BE8 373F0 F03EF 17433            RDRAMB R0,BAKFACTR            ;GET SAVED BACKING FACTOR
01BE9 29170 333EB 17434            SUBB AB,FOUTL,R7,R0 & BOZ $+2 ;IS IT < CURRENT BACKING FACTOR?
01BEA FFFFF 343F0 17435            CJMP AGB,SECSAV10             ;YES, NEXT FUNCT. IS A WRITE
01BEB 3417F F03EF 17436            ALDRAM R7,BAKFACTR            ;SAVE MIN. BACKING FACTOR (READ)
                17437   ;
                17438   ;    READ FUNCTION ON FUJI M2333K OR SWALLOW-4.
                17439   ;
01BEC 34315 FFFFF 17441            RTOR     R1,R5                ;$Sector no. to R5
01BED 1D345 F0007 17442            SUBA     DB,B,MSK1,R5 & NOP D#7 ;$Decr. sector by 7
01BEE FFFFF 77BEF 17459            CJARB    NUCZ,SECSAV9         ;IF POSITIVE NO., SECSAV4A
                17460   ;
01BEF 053F5 5FAD6 17461   SECSAV9:  ADD DB,B,RAME,R5 & JAR AMPREC ;MAKE THE SECTOR POSITIVE
                17462   ;
                17463   ;    WRITE FUNCTION ON FUJI M2333K OR SWALLOW-4.
                17464   ;
01BF0 1D345 F0007 17466   SECSAV10: SUBA DB,B,MSK1,R5 & NOP D#7  ;$DECR. SECTOR BY 7
01BF1 FFFFF 77BEF 17483            CJARB    NUCZ,SECSAV9         ;IF POSITIVE NO., SECSAV4A
                17493
                17494   TITLE SWALLOW-5 R.P.S. CODE
```

Addr Line SWALLOW-5 R.P.S. CODE

```
                    17496
01BF2 66150 F0008 17497    SECSAV20: TSTQEQU 8                          ;DRIVE TYPE 4 (SWALLOW-5)?
01BF3 37341 A3801 17498              RDREGB MSK1,R1 & CRTS NALUZ,N@1     ;NO, RETURN; PRESET SECTOR TO 1
01BF4 FFFFF E1BE6 17500              JMP SECSAV8A                        ;DON'T USE TABLE FOR HARD SECTORS
                    17501    ;
01BF5 65058 F0010 17506              TSTRMSK R8,10                       ;$RECORD LENGTH = 116?
01BF6 3409F A3FFF 17507              RTOQ R9 & CRTS NALUZ                ;$NO, RETURN; RECORD NO. TO Q
01BF7 125F0 73BF8 17512              SBA ZQ,RSBZ,,R0 & CJARB NALUZ,$+1   ;IF REC.=0, JUMP
01BF8 3400F FFFFF 17513              RTOQ R0
                    17514    ;
01BF9 0615E F1BFC 17516              ADD DQ,FOUTL,MSK2,IREG & NOP $+D#3  ;$COMPUTE THE 116 JUMP ACCORDING...
01BFA 37929 FFFFF 17520              OR DZ,FOUTU,SEQAR,IREGH             ;TO (REC.-1)/2
01BFB FFFFF 6F7FF 17521              JIR
                    17522    ;                                           ;  RECORD
01BFC 05341 5F800 17523              ADD DB,B,MSK1,R1 & JAR D#0          ; 1 OR  2
01BFD 05341 5F801 17524              ADD DB,B,MSK1,R1 & JAR D#1          ; 3 OR  4
01BFE 05341 5F803 17525              ADD DB,B,MSK1,R1 & JAR D#3          ; 5 OR  6
01BFF 05341 5F804 17526              ADD DB,B,MSK1,R1 & JAR D#4          ; 7 OR  8
01C00 05341 5F806 17527              ADD DB,B,MSK1,R1 & JAR D#6          ; 9 OR 10
01C01 05341 5F807 17528              ADD DB,B,MSK1,R1 & JAR D#7          ;11 OR 12
01C02 05341 5F809 17529              ADD DB,B,MSK1,R1 & JAR D#9          ;13 OR 14
01C03 05341 5F80B 17531              ADD DB,B,MSK1,R1 & JAR D#11         ;$ 15 OR 16
01C04 05341 5F80C 17532              ADD DB,B,MSK1,R1 & JAR D#12         ;$ 17 OR 18
01C05 05341 5F80E 17533              ADD DB,B,MSK1,R1 & JAR D#14         ;$ 19 OR 20
01C06 05341 5F80F 17534              ADD DB,B,MSK1,R1 & JAR D#15         ;$ 21 OR 22
01C07 05341 5F811 17535              ADD DB,B,MSK1,R1 & JAR D#17         ;$ 23 OR 24
01C08 05341 5F812 17536              ADD DB,B,MSK1,R1 & JAR D#18         ;$ 25 OR 26
01C09 05341 5F814 17537              ADD DB,B,MSK1,R1 & JAR D#20         ;$ 27 OR 28
01C0A 05341 5F816 17538              ADD DB,B,MSK1,R1 & JAR D#22         ;$ 29 OR 30
01C0B 05341 5F817 17539              ADD DB,B,MSK1,R1 & JAR D#23         ;$ 31 OR 32
01C0C 05341 5F819 17540              ADD DB,B,MSK1,R1 & JAR D#25         ;$ 33 OR 34
01C0D 05341 5F81A 17541              ADD DB,B,MSK1,R1 & JAR D#26         ;$ 35 OR 36
01C0E 05341 5F81C 17542              ADD DB,B,MSK1,R1 & JAR D#28         ;$ 37 OR 38
01C0F 05341 5F81E 17543              ADD DB,B,MSK1,R1 & JAR D#30         ;$ 39 OR 40
01C10 05341 5F81F 17544              ADD DB,B,MSK1,R1 & JAR D#31         ;$ 41 OR 42
01C11 05341 5F821 17545              ADD DB,B,MSK1,R1 & JAR D#33         ;$ 43 OR 44
01C12 05341 5F822 17546              ADD DB,B,MSK1,R1 & JAR D#34         ;$ 45 OR 46
01C13 05341 5F824 17547              ADD DB,B,MSK1,R1 & JAR D#36         ;$ 47 OR 48
01C14 05341 5F825 17548              ADD DB,B,MSK1,R1 & JAR D#37         ;$ 49 OR 50
01C15 05341 5F827 17549              ADD DB,B,MSK1,R1 & JAR D#39         ;$ 51 OR 52
01C16 05341 5F829 17550              ADD DB,B,MSK1,R1 & JAR D#41         ;$ 53 OR 54
01C17 05341 5F82A 17551              ADD DB,B,MSK1,R1 & JAR D#42         ;$ 55 OR 56
01C18 05341 5F82C 17552              ADD DB,B,MSK1,R1 & JAR D#44         ;$ 57 OR 58
01C19 05341 5F82D 17553              ADD DB,B,MSK1,R1 & JAR D#45         ;$ 59 OR 60
```

Addr Line SWALLOW-5 R.P.S. CODE

```
01C1A 05341 5F82F 17554              ADD DB,B,MSK1,R1 & JAR D#47       ;$ 61 OR 62
01C1B 05341 5F830 17555              ADD DB,B,MSK1,R1 & JAR D#48       ;$ 63 OR 64
01C1C 05341 5F832 17556              ADD DB,B,MSK1,R1 & JAR D#50       ;$ 65 OR 66
01C1D 05341 5F834 17557              ADD DB,B,MSK1,R1 & JAR D#52       ;$ 67 OR 68
01C1E 05341 5F835 17558              ADD DB,B,MSK1,R1 & JAR D#53       ;$ 69 OR 28
01C1F 05341 5F837 17559              ADD DB,B,MSK1,R1 & JAR D#55       ;$ 71 OR 72
01C20 05341 5F838 17560              ADD DB,B,MSK1,R1 & JAR D#56       ;$ 73 OR 74
01C21 05341 5F83A 17561              ADD DB,B,MSK1,R1 & JAR D#58       ;$ 75 OR 76
01C22 05341 5F83C 17562              ADD DB,B,MSK1,R1 & JAR D#60       ;$ 77 OR 78
01C23 05341 5F83D 17563              ADD DB,B,MSK1,R1 & JAR D#61       ;$ 79 OR 80
                  17631
                  17632    TITLE R.P.S. ON-SECTOR POLLING
```

Addr Line R.P.S. ON-SECTOR POLLING

```
                17634
                17635
                17636   ;    AFTER THE DEVICE END DELAY TIMER HAS EXPIRED, THIS ROUTINE WILL SELECT
                17637   ;ALL THE DRIVES THAT ARE ON CYLINDER, READ THE SECTOR NUMBER OF EACH, AND
                17638   ;ATTEMPT TO PRESENT A DEVICE END FOR ANY DRIVE FOUND TO BE ON THE DESIRED
                17639   ;SECTOR.
                17640
01C24 3715E F02B5 17641   SECPOL:   MOVE MSK2,IREG & NOP DRONCYL      ;GET THE DRIVE ON CYLINDER...
01C25 FFFFF 1F37C 17642             JSR SMBRDD$                       ;BITS FOR ALL DRIVES
01C26 34307 FFFFF 17643             RTOR R0,R7                        ;LSB'S TO R7
01C27 323F6 FFFFF 17644             QTOR R6                           ;MSB'S TO R6
                17645   SECPOL1:  JSR.P LOGICAL                     ;LOGICAL DRIVE TO Q
01C28 E0400 1F1F9 17645 +           SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
01C29 0615E F03C3 17646             ADD DQ,FOUTL,MSK2,IREG & NOP TRKNBR
01C2A 373D5 FFFFF 17647             RDREGB RAMI,R5                    ;GET SAVED SECTOR NO. THIS DRIVE
01C2B 45055 F0400 17648             TSTRBIT R5,VALSECT                ;VALID SECTOR NUMBER?
                17649               BOZ.P POLL18                      ;NO, SEND DEV. END
01C2C E0401 3341D 17649 +           SETPGE POLL18_SHR_H#B & BOZP POLL18
                17650   ;
01C2D 3437A 1F46A 17651             RTOR R7,R10 & JSR FASTSEL         ;SELECT A DRIVE
01C2E 433F1 1348A 17652             RESETR R1 & CJSR ALUZ,RDSECTA     ;IF GOOD STATUS, READ SECTOR
01C2F 61151 33453 17653             XOR AB,FOUTL,R5,R1 & BOZ SECPOL1A ;BAD STATUS, JUMP; CURRENT=DESIRED?
                17654               BOZ.P POLL18                      ;YES, SEND DEVICE END
01C30 E0401 3341D 17654 +           SETPGE POLL18_SHR_H#B & BOZP POLL18
                17655   ;
                17656   ;    The following new (May 1990) code will reduce time-outs on systems that
                17657   ;have many (more than 21) drives and are doing Sector Position Relative Seeks
                17658   ;in BYPASS.  This code opens a "window" on a per-drive basis to poll for a
                17659   ;larger sector range on drives that have had Device End pending for a longer
                17660   ;time than other drives.  The maximum "window" opening is 7 sectors before
                17661   ;the desired sector.  The window always starts at 0 and opens 1 sector at a
                17662   ;time approximately every disk revolution that a sector match is not made.
                17663   ;
                17668               JSR.P    LOGICAL                  ;LOGICAL DRIVE NO. TO Q
01C31 E0400 1F1F9 17668 +           SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
01C32 0615E F0474 17669             ADD DQ,FOUTL,MSK2,IREG & NOP LASTSECT ;LOC. OF LAST SECTOR READ
01C33 373D8 FFFFF 17670             RDREGB   RAMI,R8                  ;GET LAST SECTOR READ INTO R8
01C34 FFFFF 33C36 17671             BNZ      $+2                      ;IF ANY SECTOR, JUMP
01C35 3411D E1C53 17672             ALDREG   R1,RAMI & JMP SECPOL1A   ;SAVE CURRENT SECTOR AS LAST
                17673   ;
01C36 45058 F0380 17674             TSTRBIT  R8,380                   ;ISOLATE WINDOW OPENING IN Q
01C37 60388 FFFFF 17675             XOR      AQ,B,R8,R8               ;DELETE THEM FROM LAST SECTOR
01C38 29118 FFFFF 17676             SUBB     AB,FOUTL,R1,R8           ;CURRENT SECTOR < LAST?
01C39 29115 35450 17677             SUBB AB,FOUTL,R1,R5 & CJMP ALB,SECPOL1F ;YES, JUMP; CUR.> DESIRED?
01C3A 29185 3543F 17678             SUBB AB,FOUTL,R8,R5 & CJMP ALB,SECPOL1C ;NO, JUMP; LAST < DESIRED?
01C3B FFFFF 3443F 17679             CJMP     AGB,SECPOL1C             ;NO, JUMP
                17680   ;
01C3C 66150 F0380 17681   SECPOL1B: TSTQEQU  380                      ;MAX. WINDOW OPENING?
```

Addr Line R.P.S. ON-SECTOR POLLING

```
01C3D FFFFF 3343F 17682            BOZ     $+2                     ;YES, JUMP
01C3E 0605F F0080 17683            ADD     DQ,Q,MSK2 & NOP H#80    ;OPEN THE WINDOW WIDER
01C3F 3011D FFFFF 17684   SECPOL1C: OR      AQ,FOUTL,R1,RAM[        ;SAVE WINDOW & LAST SECTOR
                  17685   ;
01C40 133F5 407FF 17686   SECPOL1D: DECRB   R5 & PUSH               ;DECR. DESIRED SECTOR
01C41 45155 F0400 17687            TSTBIT  R5,VALSECT              ;POSITIVE NUMBER?
01C42 323F2 33C47 17688            QTOR    R2 & BNZ SECPOL1E       ;YES, JUMP;  WINDOW TO R2
                  17689            JSR.P   LOGICAL                 ;LOGICAL DRIVE NO. TO Q
01C43 E0400 1F1F9 17689 +          SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
01C44 0615E F0494 17690            ADD DQ,FOUTL,MSK2,IREG & NOP MAXSECT ;LOC. OF MAX. SECTOR THIS DRIVE
01C45 373D5 FFFFF 17691            RDREGB  RAM],R5                 ;GET MAX. SECTOR TO R5
01C46 3402F FFFFF 17692            RTOQ    R2                      ;WINDOW BACK TO Q-REG.
                  17693   ;
01C47 61151 FFFFF 17694   SECPOL1E: XOR     AB,FOUTL,R5,R1          ;ON-SECTOR?
                  17695            CJPOP.P ALUZ,POLL1B             ;YES, SEND DEVICE END
01C48 E0401 B341D 17695 +          SETPGE POLL1B_SHR_H#B & CJPOPP ALUZ,POLL1B
                  17696   ;
01C49 46150 F0380 17697            TESTQ   380                     ;IS THE WINDOW OPEN?
01C4A FFFFF 33453 17698            BOZ     SECPOL1A                ;IF NOT, CONTINUE
                  17699   ;
01C4B 45155 F0001 17700            TSTBIT  R5,1                    ;IS THIS AN ODD SECTOR?
01C4C 320FF 33C4E 17701            OR      ZQ,Q & BNZ $+2          ;YES, JUMP; CHECK THE WINDOW
01C4D 1E05F F0080 17702            SUBA    DQ,Q,MSK2 & NOP H#80    ;CLOSE THE WINDOW
01C4E 133F5 D37FF 17703            DECRB   R5 & CLOOP ALUZ         ;IF PARTIALLY OPEN, CHECK AGAIN
01C4F FFFFF E1C53 17704            JMP     SECPOL1A                ;OTHERWISE, CONTINUE
                  17705   ;
01C50 29185 3443C 17706   SECPOL1F: SUBB AB,FOUTL,R8,R5 & CJMP AGB,SECPOL1B ;IF CUR.> DESIRED, JUMP
01C51 FFFFF 3543C 17707            CJMP    ALB,SECPOL1B            ;IF LAST < DESIRED, JUMP
01C52 FFFFF E1C3F 17708            JMP     SECPOL1C
                  17709   ;
                  17710   ;    End of new code.
                  17711   ;
01C53 41174 FFFFF 17712   SECPOL1A: AND AB,FOUTL,R7,R4              ;IF DRIVE ON CYL. BIT IS...
01C54 41163 33C58 17713            AND AB,FOUTL,R6,R3 & BNZ SECPOL2 ;SET FOR THIS DRIVE, JUMP
01C55 FFFFF 33C58 17714            BNZ SECPOL2
01C56 37352 F02B5 17715            RDMSKB R2,DRONCYL               ;ELSE, SET THE DRIVE ON CYLINDER...
                  17716            JSR.P ADDABIT                   ;BIT FOR THIS DRIVE
01C57 E0400 1F1FF 17716 +          SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
                  17717   ;
01C58 E100B 5F465 17718   SECPOL2:  CTL HSSM,RSTUSEL & LDAR SECPOL4  ;DE-SELECT THE DRIVE
01C59 37352 F034E 17719            RDMSKB R2,SECMASK
01C5A 3412E 1F37C 17720            ALDREG R2,IREG & JSR SMBRDD$    ;GET SECTOR POLLING MASK
01C5B 51307 FFFFF 17721   SECPOL2A: MASK AB,B,R0,R7                 ;IF ANY SECTOR INTERRUPT PENDING...
01C5C 323F1 33C5F 17722            QTOR R1 & BNZ SECPOL3           ;ON DRIVE 0-23, JUMP
01C5D 51016 FFFFF 17723            MASK AB,Q,R1,R6                 ;IF NO INTERRUPT PENDING ON...
01C5E 433F0 33460 17724            RESETR R0 & BOZ $+2             ;DRIVE 24-31, JUMP
01C5F 34370 E0C45 17725   SECPOL3:  RTOR R7,R0 & JMP DRVNBR         ;COMPUTE THE DRIVE NUMBER
                  17726   ;
01C60 3412E 1F37D 17727            ALDREG R2,IREG & JSR SMBWRTD$   ;CLEAR SECTOR MASK ALL DRIVES
01C61 3714E F02B5 17728            MOVE MSK1,IREG & NOP DRONCYL
01C62 FFFFF 1F37C 17729            JSR SMBRDD$                     ;GET THE BITS FOR DRIVES...
01C63 34304 FFFFF 17730            RTOR R0,R4                      ;ON-CYLINDER SO WE WILL ONLY...
```

Addr Line R.P.S. ON-SECTOR POLLING

```
01C64 323F3 E0C1A 17731            QTOR R3 & JMP POLL17E             ;POLL DRIVES NOT YET ON-CYLINDER
                  17732    ;
                  17733    SECPOL4:  JSR.P ADDABIT                   ;ADD THE BIT FOR THIS DRIVE...
01C65 E0400 1F1FF 17733 +            SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
                  17734    ;                                          TO SECTOR POLLING MASK
01C66 FFFFF 5F468 17735              LDAR $+2
01C67 FFFFF E1C5B 17736              JMP SECPOL2A                     ;GET THE NEXT DRIVE TO POLL
                  17737              JSR.P DOUBLE                     ;DOUBLE DRIVE NO. IF NEEDED
01C68 E0403 1F125 17737 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
01C69 343A7 E1C28 17738              RTOR R10,R7 & JMP SECPOL1        ;CHECK ANOTHER DRIVE
                  17739
                  17740
                  17741    ;    THIS ROUTINE WILL DETERMINE WHICH DRIVE IS TO BE SELECTED, UPPER OR LOWER,
                  17742    ;AND SELECT IT.  THE ROUTINE WILL NOT ATTEMPT TO CORRECT ANY DRIVE ERRORS,
                  17743    ;INCLUDING BEING UNABLE TO SELECT THE DRIVE.
                  17744    ;UPON ENTRY:  R13 = PHYSICAL DRIVE NO. (LOWER).
                  17745    ;UPON EXIT:   R1  = DRIVE STATUS.
                  17746    ;             R8  = Drive flags this drive
                  17747    ;             ALU = 0 if normal drive status
                  17748    ;REGISTERS USED:  R0,R1,R8,Q & IREG.
                  17749    ;
01C6A 5535E F0800 17750    FASTSEL:  RSTRBIT R14,800                 ;RESET SECOND DRIVE FLAG
                  17751              JSR.P GETFLAG                    ;GET FLAGS FOR THIS UNIT
01C6B E0402 1F18B 17751 +            SETPGE GETFLAG_SHR_H#B & JSRP GETFLAG
01C6C 45058 F0400 17752              TSTRBIT R8,TWOFER                ;TWO DRIVES THIS UNIT?
01C6D E100B 33475 17753              CTL HSSM,RSTUSEL & BOZ FASTSEL1  ;NO, JUMP; DE-SELECT ANY DRIVE
01C6E 5535D F0001 17754              RSTRBIT R13,1                    ;CLEAR UPPER DRIVE NO. BIT
                  17755              JSR.P CYL00                      ;LOC. OF CYLINDER THIS DRIVE
01C6F E0400 1F1E9 17755 +            SETPGE CYL00_SHR_H#B & JSRP CYL00
01C70 375F1 F02D4 17756              OR DZ,RSBZ,RAME,R1 & NOP MAXCYLD ;1/2 MAX. DRIVE CYL. TO R1
01C71 2D1D1 FFFFF 17757              SUBB DB,FOUTL,RAMI,R1            ;IF CUR. CYL. IS < 1/2 MAX...
01C72 FFFFF 34C75 17758              CJMP NAGB,FASTSEL1               ;DRIVE CYL., JUMP
01C73 3535D F0001 17759              SETRBIT R13,1                    ;ELSE, INCR. DRIVE NUMBER AND...
01C74 3535E F0800 17760              SETRBIT R14,800                  ;SET SECOND DRIVE FLAG
                  17761    ;
01C75 4505D F003F 17762    FASTSEL1: TSTRBIT R13,3F                   ;ISOLATE PHYSICAL DRIVE NO. ONLY
01C76 82200 F0000 17763              WRTE HSSMA,LDDRBO & NOP 0        ;CLEAR DRIVE BUS OUT REGISTER
01C77 323F1 4F001 17764              QTOR R1 & PSHLD 1                ;DRIVE NO. TO R1
01C78 91E1F 887FF 17765              WRTL HSSM,LDDRUSEL,R1 & CLPCNT EVCNTZ ;LOAD UNIT SELECT REGISTER
01C79 E1007 FFFFF 17766              CTL HSSM,SETUSEL                 ;SELECT THE DRIVE
01C7A 433F1 4F00C 17767              RESETR R1 & PSHLD 12             ;PRESET SELECT TIMER
01C7B C1041 B3C7D 17768              RD HSSM,RDDRST1,R1 & CJPUP NALUZ,$+2 ;READ DRIVE STATUS; IF ANY, JUMP
01C7C C1041 887FF 17769              RD HSSM,RDDRST1,R1 & CLPCNT EVCNTZ ;READ STATUS & DECR. SELECT TIMER
                  17770    ;
01C7D 46340 F0030 17771              AND DW,B,MSK1,R0 & NOP H#30      ;CARD CAGE BITS (CHALLNGR) TO R0
01C7E 91E0F E1C81 17772              WRTL HSSM,LDDRUSEL,R0 & JMP $+3
01C7F 45051 F004F 17773              TSTRBIT R1,4F                    ;DON'T CHECK SEEK END & SELECTED
01C80 C1041 33C82 17774              RD HSSM,RDDRST1,R1 & BNZ $+2     ;ANY OTHER STATUS, JUMP
01C81 C1041 9847F 17775              RD HSSM,RDDRST1,R1 & CJPCNT EVCNTZ,$-2 ;CHECK AGAIN
01C82 C1041 FFFFF 17776              RD HSSM,RDDRST1,R1
01C83 55351 F0010 17777              RSTRBIT R1,WRITPROT              ;DON'T CHECK WRITE PROTECT
```

Addr Line R.P.S. ON-SECTOR POLLING

```
01C84 65141 AF0A3 17778             XOR DB,FOUTL,ASK1,R1 & RTS SKEND+SELECT+ONCYL+READY ;CHECK NORMAL...
                  17779   ;                                          ;DRIVE STATUS & RETURN
```

Addr Line R.P.S. ON-SECTOR POLLING

```
                17781
                17782   ;    THE FOLLOWING ROUTINE WILL READ THE CURRENT SECTOR NUMBER FROM A SELECTED
                17783   ;DRIVE.  UPON EXIT: R1 WILL CONTAIN THE SECTOR NUMBER PLUS A VALID BIT IF THE
                17784   ;DRIVE HAS NORMAL STATUS (SELECTED, READY, ON CYLINDER, *SEEK ERROR, *FAULT).
                17785   ;REGISTERS USED:  R0,R1,R2
                17786   ;UPON EXIT:  R1 = Sector Read + Valid sector (bit 10) if normal drive status
                17787   ;            R1 = 0 if abnormal drive status
                17788   ;            R2 = Current Drive status
                17789   ;
01C85 C1042 FFFFF 17790   RDSECT:   RD HSSM,RDDRST1,R2
01C86 C1042 FFFFF 17791            RD HSSM,RDDRST1,R2                ;READ DRIVE STATUS
01C87 55352 F0010 17792            RSTRBIT R2,WRITPROT               ;DON'T CHECK WRITE PROTECT
01C88 65152 F00A3 17793            TSTREQU R2,SKEND+SELECT+ONCYL+READY ;NORMAL STATUS?
01C89 433F1 A3FFF 17794            RESETR R1 & CRTS NALUZ            ;NO, RETURN WITH R1 = 0
                17795   ;
                17796   RDSECTA:  JSR.P LOGICAL                      ;LOGICAL DRIVE NO. TO Q
01C8A E0400 1F1F9 17796 +          SETPGE LOGICAL_SHR_H#B & JSRP LOGICAL
01C8B 0615E F04B4 17797            ADD DQ,FOUTL,MSK2,IREG & NOP SECTIME ;LOC. OF R.P.S. TIMER THIS DRIVE
01C8C 373D0 FFFFF 17798            RDREGB RAMI,R0                    ;GET R.P.S. POLLING TIMER AND...
01C8D 133F0 33C90 17799            DECRB R0 & BNZ $+3                ;DECR. IT; IF NOT 0, JUMP
01C8E 3714D F0100 17800            MOVE     MSK1,RAMI & NOP D#256    ;Preset timer = 256
01C8F 34351 AF7FF 17801            RTOR R5,R1 & RTS                  ;FORCE ON-SECTOR & RETURN
                17802   ;
01C90 3410D FFFFF 17803            ALDREG R0,RAMI                    ;SAVE DECREMENTED TIMER
01C91 E1018 4F006 17804   RDSECT1:  CTL HSSM,SETAG4 & PSHLD 6         ;TAG 4 TO READ SECTOR
01C92 C1041 887FF 17805            RD HSSM,RDDRST1,R1 & CLPCNT EVCNTZ ;READ THE SECTOR NUMBER
01C93 E1019 4F004 17806            CTL HSSM,RSTAG4 & PSHLD 4         ;RESET TAG 4 AND...
01C94 FFFFF 88494 17807            CLPCNT EVCNTZ,$                   ;DELAY 1 µsec.
01C95 E1018 4F006 17808            CTL HSSM,SETAG4 & PSHLD 6         ;SET TAG 4 AGAIN
01C96 C1040 887FF 17809            RD HSSM,RDDRST1,R0 & CLPCNT EVCNTZ ;RE-READ THE SECTOR NUMBER
01C97 E1019 FFFFF 17810            CTL HSSM,RSTAG4                   ;RESET TAG 4
01C98 61110 FFFFF 17811            XOR AB,FOUTL,R1,R0                ;THIS SECTOR = PREVIOUS SECTOR?
01C99 35341 A3400 17812            OR DB,B,MSK1,R1 & CRTS ALUZ,VALSECT ;YES, INCLUDE VALID BIT & RETURN
01C9A FFFFF E1C91 17813            JMP RDSECT1                       ;ELSE, RE-READ THE SECTOR NUMBER
                17814
                17815
                17816   TITLE FORMAT A DEFECTIVE TRACK
```

Addr Line FORMAT A DEFECTIVE TRACK

```
                17818
                17819   ILD400:  EQU $
01C9B 0F322 E1239 17823          CALL R2,ILDRUN                  ;SELECT THE DRIVE
01C9C 433F0 FFFFF 17828          RESETR R0
01C9D 0F322 E1F17 17829          CALL R2,GTDEFCYL                ;GET THE DEFECTIVE CYLINDER
01C9E 373F1 F039C 17830          RDRAMB R1,FRSTALT               ;GET FIRST ALTERNATE CYLINDER
01C9F 29101 FFFFF 17831          SUBB AB,FOUTL,R0,R1             ;DEFECTIVE CYL.< FIRST ALTERNATE?
01CA0 FFFFF 354A6 17832          CJNP ALB,ILD401                 ;YES, JUMP
01CA1 370FF F00BA 17833          RDRAMQ ILDFLG                   ;GET ILD FLAGS AND...
01CA2 3605F F0200 17834          SETQBIT 200                     ;SET DEFECTIVE ALTERNATE FLAG
01CA3 321FF F00BA 17835          QLDRAM ILDFLG
01CA4 3410F F00B9 17836          ALDRAM R0,NDCYL                 ;END CYL.= DEFECTIVE ALTERNATE
01CA5 3410F F00B5 17837          ALDRAM R0,CURCYL                ;CUR. CYL.= DEFECTIVE ALTERNATE
                17838   ;
01CA6 3535F F0010 17839   ILD401:  SETRBIT R15,10                  ;SET DEF. TRACK FLAG
01CA7 0F322 E11C7 17840          CALL R2,GTSTHED                 ;GET THE DEFECTIVE HEAD NO.
01CA8 370FF F00BA 17841   ILD401A: RDRAMQ ILDFLG                  ;GET ILD FLAGS
01CA9 4605F F0200 17842          TSTQBIT 200                     ;DEFECTIVE ALTERNATE?
01CAA FFFFF 334AD 17843          BOZ ILD402                      ;NO, FIND NEXT ALTERNATE
01CAB 3410F F00BC 17844          ALDRAM R0,NDHD                  ;END HEAD = DEFECTIVE HEAD
01CAC FFFFF E1CB0 17845          JMP ILD403                      ;FORMAT THE DEFECTIVE ALTERNATE
                17846   ;
                17847   ;    THE FOLLOWING CODE WILL FIND THE NEXT AVAILABLE ALTERNATE TRACK.
                17848   ;
01CAD 3710F F00B6 17849   ILD402:  RESETRAM CURHD                  ;FIRST ALT. HEAD IS 0
                17850            MOVSMB FRSTALT,CURCYL           ;First alt. cyl.= Current cyl.
01CAE 373F0 F039C 17850 +        RDRAMB   R0,FRSTALT
01CAF 3410F F00B5 17850 +        ALDRAM   R0,CURCYL
                17851   ;
01CB0 37352 F1CB2 17852   ILD403:  RDMSKB R2,$+2                  ;ERROR RETRY LOC.
                17853            JSR.P SAVCONT                   ;SAVE IT IN CASE OF ERROR
01CB1 E0402 1F183 17853 +        SETPGE SAVCONT_SHR_H#B & JSRP SAVCONT
01CB2 0F322 E115A 17854          CALL R2,ILDSK0                  ;SEEK THE DRIVE
01CB3 0F322 E1CFD 17855          CALL R2,HDREAD                  ;READ THE HEADER
                17856   ;
01CB4 3710F F0018 17857          RESETRAM SEN415                 ;RESET SM-RETRY COUNT
01CB5 5535E F0200 17858          RSTRBIT R14,200                 ;RESET ALTERNATE TRACK FLAG
01CB6 370FF F00BA 17859          RDRAMQ ILDFLG                   ;GET ILD FLAGS
01CB7 46140 F0200 17860          AND DQ,FOUTL,MSK1,ZERO & NOP H#200 ;DEFECTIVE ALTERNATE?
01CB8 FFFFF 334C0 17861          BOZ ILD403A                     ;NO, CONTINUE
01CB9 651F0 F00B5 17862          XOR DB,FOUTL,RAME,R0 & NOP CURCYL  ;ACTUAL CYL.= DEFECTIVE ALTERNATE?
                17863            BOZ.P ILD115J                   ;YES, FORMAT DEFECTIVE ALTERNATE
01CBA E0402 33357 17863 +        SETPGE ILD115J_SHR_H#B & BOZP ILD115J
                17864   ;
01CBB 3410F F01FB 17865          ALDRAM R0,PASCNT1               ;SAVE THE DEFECTIVE CYLINDER
01CBC 3411F F01FC 17866          ALDRAM R1,PASCNT                ;SAVE THE DEFECTIVE HEAD
01CBD 3605F F0400 17867          SETQBIT 400                     ;SET FORMAT DEFECTIVE FLAG
```

Addr Line FORMAT A DEFECTIVE TRACK

```
01CBE 321FF F00BA 17868            QLDRAM ILDFLG                    ;SAVE ILD FLAGS
01CBF FFFFF E1357 17869            JMP ILD115J                      ;FORMAT DEFECTIVE ALTERNATE
                  17870    ;
01CC0 651F0 F00B8 17871    ILD403A: XOR DB,FOUTL,RAME,R0 & NOP STCYL ;IS THIS CYL.= DEF. CYL?
01CC1 FFFFF 334D2 17872            BOZ ILD405                       ;YES, JUMP
01CC2 651F0 F00B5 17873            XOR DB,FOUTL,RAME,R0 & NOP CURCYL ;IS THIS CYLINDER = CUR. CYL?
01CC3 FFFFF 33CD7 17874            BNZ ILD406                       ;NO, THIS TRACK IS ALREADY USED
01CC4 45053 F0001 17875            TSTRBIT R3,1                     ;IS THIS A DEFECTIVE TRACK?
01CC5 FFFFF 33CD7 17876            BNZ ILD406                       ;YES, WE CAN'T USE IT
                  17877    ;
                  17878    ;    WE HAVE FOUND THE NEXT AVAILABLE ALTERNATE TRACK
                  17879    ;
01CC6 3410F F00B9 17880    ILD404: ALDRAM R0,NDCYL                  ;SAVE THE ALTERNATE CYLINDER...
01CC7 3411F F00BC 17881            ALDRAM R1,NDHD                   ;AND THE ALTERNATE HEAD
                  17882            MOVSMB STCYL,CURCYL              ;Defective cyl.= Current cyl.
01CC8 373F0 F00B8 17882 +          RDRAMB   R0,STCYL
01CC9 3410F F00B5 17882 +          ALDRAM   R0,CURCYL
                  17883            MOVSMB STHD,CURHD                ;Defective head = Current head
01CCA 373F0 F00BB 17883 +          RDRAMB   R0,STHD
01CCB 3410F F00B6 17883 +          ALDRAM   R0,CURHD
01CCC 370FF F00BA 17884            RDRAMQ ILDFLG                    ;GET THE ILD FLAGS
01CCD 46140 F0400 17885            AND DQ,FOUTL,MSK1,ZERO & NOP H#400 ;FORMAT DEFECTIVE ALREADY SET?
                  17886            BNZ.P ILD116                     ;YES, FORMAT DEFECTIVE TRACK
01CCE E0402 33B61 17886 +          SETPGE ILD116_SHR_H#8 & BNZP ILD116
01CCF 3605F F0400 17887            SETQBIT 400                      ;SET FORMAT DEFECTIVE FLAG
01CD0 321FF F00BA 17888            QLDRAM ILDFLG
01CD1 FFFFF E1357 17889            JMP ILD115J                      ;GO FORMAT THE DEFECTIVE
                  17890    ;
01CD2 651F1 F00BB 17891    ILD405: XOR DB,FOUTL,RAME,R1 & NOP STHD  ;THIS HEAD = DEFECTIVE HEAD?
01CD3 FFFFF 33CD7 17892            BNZ ILD406                       ;NO, TRY THE NEXT TRACK
01CD4 373F0 F00B5 17893            RDRAMB R0,CURCYL                 ;GET THE ALTERNATE CYL...
01CD5 373F1 F00B6 17894            RDRAMB R1,CURHD                  ;AND THE ALTERNATE HEAD
01CD6 FFFFF E1CC6 17895            JMP ILD404
                  17896    ;
01CD7 373F1 F00B6 17897    ILD406: RDRAMB R1,CURHD                  ;GET THE CURRENT HEAD
01CD8 373F0 F020D 17898            RDRAMB R0,MAXHDI                 ;GET THE MAX. HEAD
                  17899    ;
01CD9 29101 FFFFF 17900            SUBB AB,FOUTL,R0,R1              ;IS CURRENT HEAD < MAX?
01CDA FFFFF 34CDD 17901            CJMP NAGB,ILD407                 ;NO, JUMP
01CDB 0C11F F00B6 17902            ADD AZC,FOUTL,R1,RAME & NOP CURHD ;INCR. TO NEXT HEAD...
01CDC FFFFF E1CB0 17903            JMP ILD403                       ;AND TRY IT
                  17904    ;
01CDD 3710F F00B6 17905    ILD407: RESETRAM CURHD                   ;RESET THE CURRENT HEAD
01CDE 373F1 F00B5 17906            RDRAMB R1,CURCYL                 ;GET THE CURRENT CYLINDER
01CDF 373F0 F0245 17907            RDRAMB R0,MAXCYLD1               ;GET MAX. DRIVE CYLINDER
                  17908    ;
01CE0 29101 FFFFF 17909            SUBB AB,FOUTL,R0,R1              ;IS CUR. CYL.< MAX?
01CE1 FFFFF 34CE4 17910            CJMP NAGB,ILD408                 ;NO, NO MORE ALTERNATES AVAIL.
01CE2 0C11F F00B5 17911            ADD AZC,FOUTL,R1,RAME & NOP CURCYL ;INCR. TO NEXT CYLINDER AND...
01CE3 FFFFF E1CB0 17912            JMP ILD403                       ;CHECK IT
                  17913    ;
                  17914    ILD408: EQU $
```

```
Addr            Line FORMAT A DEFECTIVE TRACK

01CE4 37350 F003E 17921             RDMSKB R0,D#62
01CE5 0F322 E0E7B 17922             CALL R2,ISSCMDA1                ;MSG: 'NO ALTS. AVAIL.'
01CE6 37350 F0024 17923             RDMSKB R0,D#36
01CE7 FFFFF E1134 17924             JMP ILD28+1                     ;TERMINATE THE ILD
                  17925
                  17926
01CE8 45053 F0400 17927   ILD410:   TSTRBIT R3,400                  ;FORMAT DEFECTIVE FLAG?
01CE9 FFFFF 334F2 17928             BOZ ILD410A                     ;NO, FORMAT IS COMPLETE
01CEA 55353 F0200 17929             RSTRBIT R3,200                  ;RESET DEFECTIVE ALTERNATE
01CEB 3413F F00BA 17930             ALDRAM R3,ILDFLG
                  17931             MOVSMB PASCNT1,STCYL            ;Restore saved Defective cyl.
01CEC 373F0 F01FB 17931 +           RDRAMB   R0,PASCNT1
01CED 3410F F00B8 17931 +           ALDRAM   R0,STCYL
                  17932             MOVSMB PASCNT,STHD              ;Restore saved Defective head
01CEE 373F0 F01FC 17932 +           RDRAMB   R0,PASCNT
01CEF 3410F F00B8 17932 +           ALDRAM   R0,STHD
01CF0 3535F F0010 17933             SETRBIT R15,10                  ;SET DEFECTIVE FLAG AND...
01CF1 FFFFF E1CAD 17934             JMP ILD402                      ;FIND NEXT AVAILABLE ALTERNATE
                  17935   ;
                  17936   ILD410A:  EQU $
01CF2 34330 E1488 17945             RTOR R3,R0 & JMP ILD215         ;EXIT THE ILD
                  17947   ;
01CF3 37350 F003A 17948   ILD411:  RDMSKB R0,D#58
01CF4 0F322 E0E7B 17949             CALL R2,ISSCMDA1                ;MSG: '. ALT. TRK.= '
01CF5 373F0 F00AF 17950             RDRAMB R0,ALTCYL                ;GET THE ALTERNATE CYLINDER...
01CF6 0F322 E1270 17951             CALL R2,PRTNBR                  ;AND PRINT IT
01CF7 37350 F0022 17952             RDMSKB R0,D#34                  ;CURSOR POSITION
01CF8 3705F F000A 17953             RDMSKQ H#A
01CF9 0F322 E0E7C 17954             CALL R2,ISSCMDA                 ;POSITION THE CURSOR
01CFA 373F0 F0080 17955             RDRAMB R0,ALTHEAD               ;GET THE ALTERNATE HEAD...
01CFB 0F322 E1270 17956             CALL R2,PRTNBR                  ;AND PRINT IT
01CFC FFFFF E1135 17957             JMP ILD29                       ;EXIT THE ILD
                  17958
                  17959
                  17960   TITLE READ A HEADER
```

Addr Line READ A HEADER

```
                    17962   ;
                    17963   ;UPON ENTRY:  CURCYL = EXPECTED CYLINDER
                    17964   ;             CURHD  = EXPECTED HEAD
                    17965   ;UPON EXIT:   R0 = ACTUAL CYLINDER
                    17966   ;             R1 = ACTUAL HEAD
                    17967   ;             R3 = FLAG BYTE
                    17968   ;Registers used:  R0-R4,R11,R12,Q & IREG.
                    17969   ;
                    17970   HDREAD:   JSR.P SAVRET5                     ;SAVE RETURN LOC.
01CFD E0402 1F2F9 17970 +             SETPGE SAVRET5_SHR_H#B & JSRP SAVRET5
01CFE 0F322 E1239 17971               CALL R2,ILDRUN                    ;SELECT THE DRIVE IF NECESSARY
01CFF 37352 F1D01 17972               RDMSKB R2,$+2                     ;SAVE RETRY-ON-ERROR RETURN...
                    17973             JSR.P SAVCONT                     ;IN CASE OF ERROR
01D00 E0402 1F183 17973 +             SETPGE SAVCONT_SHR_H#B & JSRP SAVCONT
                    17974   ;
01D01 3710F F0018 17975               RESETRAM SEN415                   ;RESET SM-RETRY COUNT
01D02 373FB F00B5 17976   HDREAD0:   RDRAMB R11,CURCYL                 ;GET THE CURRENT CYLINDER
01D03 373FC F00B6 17977               RDRAMB R12,CURHD                  ;GET THE CURRENT HEAD
01D04 37750 F8000 17978               OR DZ,LSBZ,MSK2,R0 & NOP H#8000
01D05 3130E FFFFF 17979               OR AB,B,R0,R14                    ;SET FIND ALTERNATE FLAG
01D06 37350 F1D02 17980               RDMSKB R0,HDREAD0                 ;SAVE SM-ERROR RETURN...
                    17981             JSR.P SAVSMRTN                    ;IN CASE OF ERROR
01D07 E0400 1F4C6 17981 +             SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
                    17982   ;
01D08 343B3 1F569 17983   HDREAD1:  RTOR R11,R3 & JSR DRVHDR            ;CYL. TO R3 & LOAD EXPECTED HDR.
01D09 B1F00 F0030 17984               WRTE HSSM,LDXMCJ & NOP RDHDR
01D0A E100D FFFFF 17985               CTL HSSM,XMCJCMD                  ;START READ HEADER
                    17986   ;
01D0B 35355 F0100 17987               SETRBIT R5,100                    ;SET SM-READ ACTIVE
01D0C 3535F F0001 17988               SETRBIT R15,1                     ;SET READ HEADER FLAG
01D0D 37350 F1D14 17989               RDMSKB R0,HDREAD3                 ;SM-DONE RETURN LOC.
                    17990             JSR.P SAVIRPT                     ;SAVE IT
01D0E E0400 1F4C0 17990 +             SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
01D0F 37350 F61A8 17991               RDMSKB R0,61A8                    ;SET 50-MS. TIMER
                    17992             JSR.P SAVSMTIM                    ;SAVE IT
01D10 E0400 1F4C2 17992 +             SETPGE SAVSMTIM_SHR_H#B & JSRP SAVSMTIM
                    17993   ;
                    17994             JSR.P SMIRPT                      ;CHECK FOR AN INTERRUPT
01D11 E0400 1F37C 17994 +             SETPGE SMIRPT_SHR_H#B & JSRP SMIRPT
                    17995             JSR.P SMTIMER                     ;DECR. THE TIMER...
01D12 E0400 1F4DB 17995 +             SETPGE SMTIMER_SHR_H#B & JSRP SMTIMER
01D13 FFFFF E1D11 17996               JMP $-2                           ;AND CHECK AGAIN
                    17997   ;
01D14 E1000 FFFFF 17998   HDREAD3:  CTL HSSM,DIFRST
                    17999             JSR.P HSWAITR                     ;MPC WAIT LOOP ROUTINE
01D15 E0400 1F525 17999 +             SETPGE HSWAITR_SHR_H#B & JSRP HSWAITR
```

```
 Addr            Line READ A HEADER

01D16 37750 F8000 18000            OR DZ,LSBZ,MSK2,R0 & NOP H#8000
01D17 5130E FFFFF 18001            MASK AB,B,R0,R14                  ;RESET FIND ALT. FLAG
01D18 5535F F0401 18002            RSTRBIT R15,401                   ;RESET HDR. READ & SM-ERROR FLAGS
01D19 55355 F0100 18003            RSTRBIT R5,100                    ;RESET SM-ACTIVE FLAG
                  18004    ;
                  18005             JSR.P READCYL                    ;READ THE CYLINDER NO.
01D1A E0402 1F579 18005 +          SETPGE READCYL_SHR_H#B & JSRP READCYL
01D1B C10B1 FFFFF 18006            RD HSSM,RDDRHD,R1
01D1C C10B1 C0001 18007            RD HSSM,RDDRHD,R1 & LDCT 1        ;READ THE HEAD NO. INTO R1 AND...
01D1D 45351 F001F 18008            MSKRBIT R1,1F                     ;DELETE THE DRIVE NO. MSB'S
01D1E C1083 9851E 18009            RD HSSM,RDDRFLG,R3 & CJPCNT EVCNTZ,$ ;READ HEADER FLAGS INTO R3
01D1F 37160 E1306 18010            MOVE SMCST,ZERO & JMP RTRN5       ;RESET CT-BUS PARITY & RETURN
                  18011
                  18012
```

Addr Line READ A HEADER

```
18014
18015
18016   TITLE HSSM DMA ADDRESS CHECKING
```

Addr Line HSSM DMA ADDRESS CHECKING

```
                18018
                18019   ;               DISPLAY 51C-----------------------
                18020   ;DMABADU:  EQU H#354          ;LAST SH DMA ADDR. UPPER IF ERROR
                18021   ;DMABADL:  EQU H#355          ;LAST SH DMA ADDR. LOWER IF ERROR
                18022   ;SPARE:    EQU H#356
                18023
                18024   ;               DISPLAY 51D-----------------------
                18025   ;DMAGOODU: EQU H#357          ;EXPECTED DMA ADDR. UPPER IF ERROR
                18026   ;DMAGOODL: EQU H#358          ;EXPECTED DMA ADDR. LOWER IF ERROR
                18027   ;OVCOPY:   EQU H#359          ;ONLY VALID COPY ERROR FLAG
                18028
01D20 4505F F8202 18029   DMACHK:   TSTRBIT R15,8202                ;NO DATA XFER./TRUNC./ECC CORR?
01D21 420FF A3FFF 18030             RESETQ & CRTS NALUZ             ;YES, RETURN NOW
01D22 4505E F0108 18031             TSTRBIT R14,108                 ;READ HEADER OR H.A. FLAGS?
01D23 420FF A3FFF 18032             RESETQ & CRTS NALUZ             ;YES, DMA WAS NOT USED
                18037   DMACHKP1: JSR.P CHKBUSY                   ;Check for HSSM busy
01D24 E0400 1F535 18037 +           SETPGE CHKBUSY_SHR_H#B & JSRP CHKBUSY
01D25 C10A2 FFFFF 18038             RD HSSM,RDDRCYLH,R2             ;READ THE LAST SM...
01D26 C10A2 4F001 18039             RD HSSM,RDDRCYLH,R2 & PSHLD 1   ;DMA ADDRESS UPPER...
01D27 C1091 887FF 18040             RD HSSM,RDDRCYLL,R1 & CLPCNT EVCNTZ ;AND LOWER
01D28 C2000 FFFFF 18041             RD HSSMA,RDINTR,R0              ;Read HSSM interrupts
01D29 45050 F0010 18042             TSTRBIT R0,10                   ;Did the HSSM go 'busy'?
01D2A FFFFF 33D24 18043             BNZ DMACHKP1                    ;Yes, wait for not 'busy'
01D2B 370FF F0352 18044             RDRAMQ NEXTSML                  ;Get expected DMA lsb's
                18045   ;
01D2C 45155 F0800 18046             TSTBIT R5,800                   ;WRITE OPERATION?
01D2D FFFFF 3352F 18047             BOZ $+2                         ;IF NOT, JUMP
01D2E 0A0FF FFFFF 18048             INCRQ                           ;INCR. LSB'S +1
01D2F 4605F F00FF 18049             TSTQBIT H#FF                    ;ISOLATE VALID BITS
01D30 651F2 F0351 18050             XOR DB,FOUTL,RAME,R2 & NOP NEXTSMU ;UPPER DMA ADDR. O.K?
01D31 60110 33D34 18051             XOR AQ,FOUTL,R1,ZERO & BNZ $+3  ;NO, JUMP; LOWER ADDR. O.K?
01D32 FFFFF 33D34 18052             BNZ $+2                         ;NO, JUMP
01D33 420FF AF7FF 18053             RESETQ & RTS                    ;ADDRESS IS GOOD, RETURN
                18054   ;
01D34 3412F F0354 18055             ALDRAM R2,DMABADU               ;SAVE THE BAD ADDR. MSB'S...
01D35 3411F F0355 18056             ALDRAM R1,DMABADL               ;AND LSB'S FOR DISPLAY
01D36 321FF F0358 18057             QLDRAM DMAGOODL                 ;SAVE EXPECTED ADDR. LSB'S
01D37 370FF F0351 18058             RDRAMQ NEXTSMU                  ;GET EXPECTED ADDR. MSB'S AND...
01D38 321FF F0357 18059             QLDRAM DMAGOODU                 ;SAVE THEM FOR DISPLAY
                18060   ;
01D39 3705F F0008 18061             RDMSKQ 8                        ;SET SENSE 9, FORMAT 2 =...
01D3A 321FF AF021 18062             QLDREG RAME & RTS SEN209        ;MDB SM DMA CHECK & RETURN
```

Addr Line HSSM DMA ADDRESS CHECKING

```
                    18064
                    18065   ;               DISPLAY 51E-----------------------
                    18066   ;UDMBADU:  EQU H#35A          ;LAST U-DMA ADDR. UPPER IF ERROR
                    18067   ;UDMBADL:  EQU H#35B          ;LAST U-DMA ADDR. LOWER IF ERROR
                    18068   ;URDDMAU:  EQU H#35C          ;ENDING U-DMA UPPER (READ)
                    18069
                    18070   ;               DISPLAY 51F-----------------------
                    18071   ;UDMGOODU: EQU H#35D          ;EXPECTED U-DMA UPPER IF ERROR
                    18072   ;UDMGOODL: EQU H#35E          ;EXPECTED U-DMA LOWER IF ERROR
                    18073   ;URDDMAL:  EQU H#35F          ;ENDING U-DMA LOWER (READ)
                    18074
                    18075   ;    THIS ROUTINE CHECKS THE U-SIDE DMA ADDRESS AT THE COMPLETION OF A U-SIDE
                    18076   ;OPERATION.  IF THE ACTUAL AND EXPECTED ADDRESSES MISCOMPARE, THE ADDRESSES
                    18077   ;WILL BE SAVED IN THE ABOVE DISPLAYS.  IN ADDITION, BIT 6 OF SENSE BYTE 8,
                    18078   ;FORMAT 2 WILL BE SET AND A FLAG BIT (R14=1000) WILL BE SET.  SETTING THE FLAG
                    18079   ;ALLOWS THE CURRENT OPERATION TO CONTINUE, BUT AN EQUIPMENT CHECK WILL BE
                    18080   ;GENERATED AT THE END.
                    18081   ;
                    18082   CHKUDMA:  EQU $
01D3B 4505F F0002   18087             TSTRBIT R15,2                    ;Truncation?
01D3C FFFFF A3FFF   18088             CRTS NALUZ                       ;Yes, return
01D3D 45055 F0800   18089             TSTRBIT R5,800                   ;Write operation?
01D3E FFFFF 33D42   18090             BNZ CHKUDMA1                     ;Yes, jump
                    18091   ;
01D3F 373F2 F035C   18092             RDRAMB R2,URDDMAU                ;GET EXPECTED READ DMA UPPER...
01D40 373F1 F035F   18093             RDRAMB R1,URDDMAL                ;AND LOWER
01D41 FFFFF E1D44   18094             JMP CHKUDMA2
                    18095   ;
01D42 373F2 F0268   18096   CHKUDMA1: RDRAMB R2,UADDRU                 ;GET EXPECTED WRITE DMA UPPER...
01D43 373F1 F0269   18097             RDRAMB R1,UADDRL                 ;AND LOWER
                    18098   ;
01D44 45055 F0003   18099   CHKUDMA2: TSTRBIT R5,3                     ;ODD START OR ODD END FLAG?
01D45 60355 3354A   18100             XOR AQ,B,R5,R5 & BOZ CHKUDMA3    ;YES, RESET IT; NO, JUMP
01D46 0B3F1 FFFFF   18101             INCRB R1                         ;INCR. EXPECTED DMA ADDR.
01D47 45341 F00FF   18102             AND DB,B,MSK1,R1 & NOP H#FF      ;ISOLATE 8 LSB'S
01D48 FFFFF 33D4A   18103             BNZ CHKUDMA3                     ;IF THEY ARE NOT 0, JUMP
01D49 0B3F2 FFFFF   18104             INCRB R2                         ;INCR. THE MSB'S
                    18105   ;
                    18106   CHKUDMA3: JSR.P CHKBUSY                    ;CHECK FOR HSSM BUSY
01D4A E0400 1F535   18106 +           SETPGE CHKBUSY_SHR_H#8 & JSRP CHKBUSY
01D4B C1030 FFFFF   18107             RD HSSM,RDUARM,R0                ;GET U-SIDE DMA MSB'S...
01D4C C1030 4F001   18108             RD HSSM,RDUARM,R0 & PSHLD 1      ;INTO R0 AND...
01D4D C1023 887FF   18109             RD HSSM,RDUARL,R3 & CLPCNT EVCNTZ ;THE LSB'S INTO R3
01D4E C2004 FFFFF   18110             RD HSSMA,RDINTR,R4
01D4F 40140 FFFFF   18111             AND AQ,FOUTL,R4,ZERO             ;CHECK AGAIN FOR BUSY
01D50 FFFFF 33D4A   18112             BNZ CHKUDMA3                     ;IF YES, WAIT FOR NOT BUSY
                    18113   ;
01D51 61120 FFFFF   18114             XOR AB,FOUTL,R2,R0               ;MSB'S O.K?
01D52 61131 33D55   18115             XOR AB,FOUTL,R3,R1 & BNZ CHKUDMA5 ;NO, JUMP; LSB'S O.K?
01D53 FFFFF 33D55   18116             BNZ CHKUDMA5                     ;NO, JUMP
```

Addr Line HSSM DMA ADDRESS CHECKING

```
01D54 FFFFF AF7FF 18118             RTS                             ;$Otherwise, normal return
                  18122     ;
01D55 4505E F1000 18123     CHKUDMA5: TSTRBIT R14,1000                ;IS THIS THE FIRST ERROR?
01D56 FFFFF A3FFF 18125             CRTS    NALUZ                   ;$If not, return
01D57 3412F F035D 18129             ALDRAM R2,UDMGOODU              ;SAVE EXPECTED U-DMA UPPER...
01D58 3411F F035E 18130             ALDRAM R1,UDMGOODL              ;AND LOWER FOR DISPLAY
01D59 3410F F035A 18131             ALDRAM R0,UDMBADU               ;SAVE BAD U-DMA ADDR. UPPER...
01D5A 3413F F0358 18132             ALDRAM R3,UDMBADL               ;AND LOWER FOR DISPLAY
01D5B 3705F F0002 18133             RDMSKQ 2                        ;MDR U-SIDE DMA CHECK SENSE
01D5C 321FF F0020 18134             QLDRAM SEN208                   ;SET SENSE 8, FORMAT 2
01D5D 3535E F1000 18135             SETRBIT R14,1000                ;SET THE ERROR FLAG
01D5E FFFFF AF7FF 18137             RTS                             ;$and return
                  18160
                  18161
                  18162     ;    THIS ROUTINE SAVES THE EXPECTED U-DMA ADDRESS ON READS.
                  18163     ;
01D5F 45155 F4000 18164     NXTUDMA:  TSTBIT R5,4000                  ;ADD ONE BYTE FLAG?
01D60 FFFFF 33566 18165             BUZ NXTUDMA1                    ;IF NOT, JUMP
01D61 55355 F4000 18166             RSTRBIT R5,4000                 ;RESET ADD ONE BYTE FLAG
01D62 0A0FF FFFFF 18167             INCRQ                           ;INCR. EXPECTED DMA LSB'S
01D63 4605F F00FF 18168             TSTQBIT H#FF                    ;ISOLATE 8 LSB'S
01D64 FFFFF 33D66 18169             BNZ NXTUDMA1                    ;IF THEY AREN'T 0, JUMP
01D65 0B3F1 FFFFF 18170             INCRB R1                        ;ELSE, INCR. THE MSB'S
01D66 3491F F035C 18171     NXTUDMA1: ALDRAMU R1,URDDMAU              ;SAVE EXPECTED U-DMA UPPER...
01D67 321FF AF35F 18172             QLDREG RAME & RTS URDDMAL       ;AND LOWER; RETURN
                  18173
                  18174
                  18175     TITLE MORE MISCELLANEOUS ROUTINES
```

```
 Addr             Line MORE MISCELLANEOUS ROUTINES


                  18177
01D68 FFFFF E02C0 18178    MULTPLYD: JMP MULTPLY
                  18179
                  18180
                  18181    ;    THIS ROUTINE INCLUDES THE LOGICAL DRIVE NUMBER IN THE HEADER THAT IS
                  18182    ;WRITTEN ON EACH RECORD OF A DRIVE.  THE 4  LSB'S OF DRIVE NUMBER ARE WRITTEN
                  18183    ;IN THE 4 MSB'S OF THE UPPER CYLINDER BYTE.  THE 2 MSB'S OF DRIVE NUMBER ARE
                  18184    ;WRITTEN IN THE 3 MSB'S OF THE HEAD BYTE.
                  18185    ;
                  18186    ;UPON ENTRY:  R3  = LOGICAL CYLINDER
                  18187    ;             R9  = RECORD NO.
                  18188    ;             R12 = HEAD NO.
                  18189    ;             R13 = PHYSICAL DRIVE NO.
                  18190    ;REGISTERS USED:  R0,R1,R3 & Q.
                  18191    ;
01D69 433F0 FFFFF 18197    DRVHDR:   RESETR R0                          ;NO DRIVE NO. IN HEADER OPTION
01D6A 45050 F0060 18199              TSTRBIT R0,60                      ;ISOLATE THE 2 MSB'S IN Q
01D6B 60700 FFFFF 18200              XOR AQ,LSBZ,R0,R0                  ;ISOLATE 4 LSB'S IN R0
01D6C 303C1 FFFFF 18201              OR AQ,B,R12,R1                     ;MERGE MSB'S WITH HEAD NO.
01D6D 91C1F 4F004 18202              WRTL HSSM,LDDRHD,R1 & PSHLD 4      ;LOAD HEAD NO. & DRIVE MSB'S
                  18203              LSHFT2 R0,R0                       ;POSITION DRIVE LSB'S AND
01D6E 01700 887FF 18203 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01D6F 3705F F00FF 18212              RDMSKQ H#FF
01D70 31303 FFFFF 18213              OR AB,B,R0,R3                      ;MERGE DRIVE LSB'S WITH CYL. MSB'S
01D71 40330 FFFFF 18214              AND AQ,B,R3,R0                     ;ISOLATE CYL. LSB'S IN R0...
01D72 91A0F 4F001 18215              WRTL HSSM,LDDRCYL,R0 & PSHLD 1     ;LOAD THEM
                  18216              LSHFT2 R3,R3                       ;POSITION DRIVE LSB'S/CYL. MSB'S...
01D73 01733 887FF 18216 +            ADD AB,LSBZ,R3,R3 & CLPCNT EVCNTZ
01D74 A1B3F FFFFF 18217              WRTU HSSM,LDDRCYM,R3               ;AND LOAD THEM
01D75 91D9F AF7FF 18223              WRTL HSSM,LDDRREC,R9 & RTS         ;LOAD THE RECORD NO. & RETURN
```

```
 Addr             Line MORE MISCELLANEOUS ROUTINES


                  18230
                  18231
01D76 E0090 FFFFF 18232    UPITGOOD: CTL SMC,UMPRST                    ;DE-SELECT HAIO
01D77 E407E 1F581 18233              CTL HIO,SPILCH1 & JSR CLRHIO      ;SELECT HAIO-1 & CLEAR E.F.
01D78 E407D 1F581 18234              CTL HIO,SPILCH2 & JSR CLRHIO      ;SELECT HAIO-2 & CLEAR E.F.
01D79 E407B 1F581 18235              CTL HIO,SPILCH3 & JSR CLRHIO      ;SELECT HAIO-3 & CLEAR E.F.
01D7A E4077 1F581 18236              CTL HIO,SPILCH4 & JSR CLRHIO      ;SELECT HAIO-4 & CLEAR E.F.
01D7B FFFFF 3E57D 18237              CJMP CACHE,$+2                    ;If Cache, Toggle is already set
01D7C 37357 F0007 18238              RDMSKB R7,7                       ;SET TOGGLE = 7
01D7D E100B FFFFF 18239              CTL      HSSM,RSTUSEL             ;Clear Unit Select tag
01D7E E1012 FFFFF 18240              CTL      HSSM,RSTAG1              ;Clear drive Tag 1
01D7F E1013 FFFFF 18241              CTL      HSSM,RSTAG2              ;Clear drive Tag 2
01D80 FFFFF E0340 18242              JMP CLRHSSM                       ;CLEAR HSSM BOARDS & RETURN
                  18243    ;
01D81 E4404 1F584 18244    CLRHIO:   CTL HIO,HIOTGTS & JSR CLREF+1
01D82 E0090 AF7FF 18245              CTL SMC,UMPRST & RTS
                  18246    ;
01D83 E4404 FFFFF 18247    CLREF:    CTL HIO,HIOTGTS                   ;THIS ROUTINE WILL...
01D84 E4208 FFFFF 18248              CTL HIO,EFREQS                    ;RESET ANY E.F. THAT...
01D85 E4200 FFFFF 18249              CTL HIO,EFREQR                    ;MAY BE SET IN AN HAIO
01D86 E4408 AF7FF 18250              CTL HIO,HIOTGTR & RTS
                  18278
                  18279
                  18280    TITLE RELATIVE ADDRESS CONVERSION CHECKING
```

Addr Line RELATIVE ADDRESS CONVERSION CHECKING

```
                    18282
01D87 373F1 F0006 18284    ADDRCHK:  RDRAMB R1,SEN06                  ;<GET CYLINDER MSB'S & HEAD
01D88 45051 F001F 18288              TSTRBIT R1,1F                    ;ISOLATE HEAD NO. IN Q AND...
01D89 323F8 FFFFF 18289              QTOR R8                          ;MOVE IT TO R8
01D8A 60711 FFFFF 18290              XOR AQ,LSBZ,R1,R1                ;ISOLATE CYLINDER MSB'S AND...
01D8B 01711 FFFFF 18291              ADD AB,LSBZ,R1,R1                ;MOVE THEN INTO POSITION
01D8C 353F1 F0005 18292              OR DB,B,RAME,R1 & NOP SEN05      ;MERGE WITH THE CYLINDER LSB'S
                    18293    ;
                    18295              LDWD     R2,CYLDIVAM             ;< ¼ Cylinder divisor
01D8D 373F2 F02D7 18295 +            RDRAMB   R2,CYLDIVAM               ;GET 12 MSBS
01D8E D0FF2 FFFFF 18295 +            RDXFR    SMC,,R2                 ;PUT IN UPPER REGISTER
01D8F 353F2 F02D8 18295 +            OR DB,B,RAME,R2 & NOP CYLDIVAM+1   ;OR IN 12 LSBS
01D90 01722 1F568 18299              ADD AB,LSBZ,R2,R2 & JSR MULTPLYD$  ;COMPUTE WORD RELATIVE CYLINDER
01D91 323F4 FFFFF 18300              QTOR R4                          ;MSB'S TO R4
01D92 34303 FFFFF 18301              RTOR R0,R3                       ;LSB'S TO R3
                    18302    ;
01D93 377F2 F02D9 18304              OR DZ,LSBZ,RAME,R2 & NOP WRDSTRK ;< 1/4 WORDS PER TRACK TIMES 2...
01D94 337F2 FFFFF 18308              OR ZB,LSBZ,,R2                   ;TIMES 2 AGAIN, TIMES...
01D95 34381 1F5A4 18309              RTOR R8,R1 & JSR ADDRCHKA$       ;HEAD NO.= FILE WORD RELATIVE HEAD
                    18310    ;
01D96 373F2 F02CD 18311              RDRAMB R2,RECLNTH                ;RECORD LENGTH TIMES...
01D97 273F1 F0015 18313              SAB DZ,B,RAME,R1 & NOP SEN412    ;<RECORD NUMBER -1
01D98 FFFFF 1F5A4 18317              JSR ADDRCHKA$                    ;COMPUTE FILE WORD RELATIVE RECORD
                    18318    ;
01D99 01363 FFFFF 18319              ADD AB,B,R6,R3                   ;ADD WORD NUMBER
01D9A FFFFF 31D9C 18320              CJMP NCARRY,$+2                  ;IF NO CARRY, JUMP
01D9B 0B3F4 FFFFF 18321              INCRB R4                         ;INCR. MSB'S
                    18322    ;
01D9C 3715E F0288 18323              MOVE MSK2,1REG & NOP EFADDRES
01D9D FFFFF 1F37C 18324              JSR SMBRDD$                      ;GET SAVED RELATIVE ADDRESS
01D9E 61130 FFFFF 18325              XOR AB,FOUTL,R3,R0               ;LSB'S THE SAME?
01D9F 60140 33DA1 18326              XOR AQ,FOUTL,R4,ZERO & BNZ $+2   ;NO, JUMP; MSB'S THE SAME?
01DA0 FFFFF A37FF 18327              CRTS ALUZ                        ;YES, RETURN
                    18328    ;
                    18329    ;   ADDRESS MISCOMPARE.  POSSIBLE ALU FAILURE OR SMB MEMORY FAILURE.
                    18330    ;
                    18331              SETMBIT 80,SEN209              ;Set Sense = ALU failure
01DA1 37350 F0080 18331 +            RDMSKB   R0,80
01DA2 3410F F0021 18331 +            ALDRAM   R0,SEN209
01DA3 FFFFF E04AD 18332              JMP READ62                       ;SEND EQUIP. CHECK
```

Addr Line RELATIVE ADDRESS CONVERSION CHECKING

```
                18333   ;
01DA4 FFFFF 1F568 18334   ADDRCHKA: JSR MULTPLYD$
01DA5 01303 FFFFF 18335           ADD AB,B,R0,R3                ;ADD THE LSB'S
01DA6 FFFFF A1FFF 18336           CRTS NCARRY                   ;IF NO CARRY, RETURN
01DA7 083F4 AF7FF 18337           INCRB R4 & RTS                ;INCR. MSB'S AND RETURN
                18338
                18339
                18340   TITLE HSSM DATA PATH CONFIDENCE TEST
```

Addr Line HSSM DATA PATH CONFIDENCE TEST

```
                18342    ;    THIS ROUTINE IS THE HSSM DATA PATH CONFIDENCE TEST.  IT WORKS IN
                18343    ;CONJUNCTION WITH THE MPC TO:
                18344    ;   1.  DOUBLE CHECK THE DRIVE POSITION. I.E., CORRECT CYLINDER & HEAD
                18345    ;   2.  CHECK THE XMC1 BYTE-TO-WORD CONVERSION.
                18346    ;   3.  CHECK THE XMC1 WORD-TO-BYTE CONVERSION.
                18347    ;   4.  CHECK DATA THRU THE MDB.
                18348    ;   5.  CHECK MDB SEQUENCER OPERATION (READ & WRITE DATA).
                18349    ;   6.  CHECK PROPER DMA OPERATION (SM & U-SIDE)
                18350    ;   7.  CHECK DIF FIFO (READ HEADERS).
                18351    ;
                18352    ;    THIS CONFIDENCE TEST WILL BE EXECUTED PRIOR TO THE READ OR WRITE OF THE
                18353    ;FIRST RECORD OF A DATA TRANSFER.  NORMAL ERROR RECOVERY PROCEDURES WILL BE
                18354    ;ATTEMPTED WHEN READING HEADERS.  HOWEVER, NO ERROR RECOVERY WILL BE ATTEMPTED
                18355    ;IF ANY ERROR IS ENCOUNTERED WHILE EXECUTING THE CONFIDENCE TEST.
                18356    ;
                18357    ;UPON ENTRY:  R11 = CURRENT CYLINDER (LOGICAL)
                18358    ;             R12 = CURRENT HEAD
                18359    ;Registers used: R0,R1,R3-R5,R8,Q & IREG.
                18360    ;
                18361    BYTE2WD:  EQU H#2C                          ;MPC BYTE-TO-WORD ROUTINE
                18362    WD2BYTE:  EQU H#2D                          ;MPC WORD-TO-BYTE ROUTINE
                18363    ;
                18364    PATHCHK:  EQU $
01DA8 3710F F0018 18380             RESETRAM SEN415                 ;RESET SM-RETRY COUNT
                18381               SETABIT  40,SEN109              ;HSSM Test failure Sense
01DA9 37350 F0040 18381 +           RDMSKB   R0,40
01DAA 3410F F0009 18381 +           ALDRAM   R0,SEN109
01DAB 3410F F0021 18382             ALDRAM   R0,SEN209              ;To Sense 9, Format 2 also
                18383    ;
01DAC 37350 F1D08 18384             RDMSKB R0,HDREAD1               ;READ HEADER RETRY ON...
                18385               JSR.P SAVSMRTN                  ;ERROR RETURN LOC.
01DAD E0400 1F4C6 18385 +           SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
01DAE 37352 F1DB1 18386             RDMSKB R2,$+D#3                 ;NORMAL RETURN LOC.
                18387               JSR.P SAVRET5                   ;SAVE IT AND...
01DAF E0402 1F2F9 18387 +           SETPGE SAVRET5_SHR_H#B & JSRP SAVRET5
01DB0 E46F0 E1D08 18388             CTL HIO,DENCHT & JMP HDREAD1    ;READ A HEADER; DISABLE HAID TAGS
                18389    ;
01DB1 373F2 F0018 18390             RDRAMB R2,SEN415                ;GET THE RETRY COUNT
01DB2 611C1 FFFFF 18391             XOR AB,FOUTL,R12,R1             ;ACTUAL HEAD = EXPECTED HEAD?
                18392               CJSR.P NALUZ,SMERR10            ;NO, ATTEMPT ERROR RECOVERY
01DB3 E0400 13E18 18392 +           SETPGE SMERR10_SHR_H#B & CJSRP NALUZ,SMERR10
01DB4 343B4 FFFFF 18400             RTOR R11,R4                     ;EXPECTED CYL. TO R4
01DB5 61140 FFFFF 18402             XOR AB,FOUTL,R4,R0              ;ACTUAL CYL.= EXPECTED CYL?
                18403               CJSR.P NALUZ,SMERR11            ;NO, ATTEMPT ERROR RECOVERY
01DB6 E0400 13E27 18403 +           SETPGE SMERR11_SHR_H#B & CJSRP NALUZ,SMERR11
```

Addr Line HSSM DATA PATH CONFIDENCE TEST

```
01DB7 433FA FFFFF 18404            RESETR R10                      ;CLEAR SECOND TEST FLAG
01DB8 3710F F00I8 18405            RESETRAM SEN415                 ;RESET SM-RETRY COUNT
                  18406    PATHCHK0: JSR.P GETFLAG                   ;GET EXPECTED HEADER FLAG INTO R8
01DB9 E0402 1F18B 18406 +          SETPGE GETFLAG_SHR_H#B & JSRP GETFLAG
01DBA 333FA FFFFF 18417            OR ZB,B,,R10
                  18418    ;
01DBB FFFFF 33DBF 18419    PATHCHK1: BNZ PATHCK1A$                   ;IF SECOND TEST FLAG, JUMP
01DBC 45058 F00FF 18420            TSTRBIT R8,H#FF                 ;ISOLATE SIGNIFICANT BITS IN Q
01DBD 60130 FFFFF 18421            XOR AQ,FOUTL,R3,ZERO            ;ACTUAL FLAGS = EXPECTED FLAGS?
01DBE E45FF 33E21 18422            CTL HIO,CLRHSIM & BNZ PATHERR2$ ;NO, ERROR EXIT
                  18423    ;
                  18424    ;    THE DRIVE IS POSITIONED PROPERLY.  THIS CODE WILL VERIFY THE PASSING OF
                  18425    ;DATA FROM THE DIF & MPC THROUGH THE MDB & PROPER ASSEMBLY OF THE DATA
                  18426    ;(BYTES) INTO WORDS ON THE XMCI.
                  18427    ;
01DBF B1F00 F002C 18428    PATHCK1A: WRTE HSSM,LDXMCJ & NOP BYTE2WD
01DC0 E100D FFFFF 18429            CTL HSSM,XMCJCMD                ;INITIATE BYTE-TO-WORD CHECK
01DC1 37350 F0D03 18430            RDMSKB R0,H#D03                 ;10-MS. TIMER
                  18431            JSR.P SAVUTIM                   ;SAVE IT IN CASE OF TIME-OUT
01DC2 E0400 1F4C4 18431 +          SETPGE SAVUTIM_SHR_H#B & JSRP SAVUTIM
01DC3 37350 F1E27 18432            RDMSKB R0,PATHERR5              ;ERROR RETURN
                  18433            JSR.P SAVSMRTN                  ;SAVE IT JUST IN CASE
01DC4 E0400 1F4C6 18433 +          SETPGE SAVSMRTN_SHR_H#B & JSRP SAVSMRTN
01DC5 37350 F1E12 18434            RDMSKB R0,PATHERR0              ;SM-DONE (ERROR) RETURN
                  18435            JSR.P SAVIRPT                   ;SAVE IT IN CASE OF ERROR
01DC6 E0400 1F4C0 18435 +          SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
01DC7 37160 FFFFF 18436            MOVE SMCST,ZERO
01DC8 FFFFF 1F630 18437            JSR ASMWRDS                     ;ASSEMBLE REFERENCE WORDS
                  18438    ;
                  18439    ;    THIS CODE WILL READ 4 WORDS FROM THE XMCI VIA THE U-BUS AND COMPARE THEM
                  18440    ;WITH THE EXPECTED VALUES THAT WERE PREVIOUSLY STORED.
                  18441    ;
01DC9 433F8 FFFFF 18442            RESETR   R8                     ;Reset data error count
01DCA 3715E F03EF 18443            MOVE MSK2,IREG & NOP WORD0-1    ;LOC. OF REFERENCE DATA
01DCB 37352 F0004 18444            RDMSKB   R2,4                   ;No. of words to check
                  18445    ;
01DCC 37354 F0004 18446    PATHCHK2: RDMSKB R4,4                     ;SET IDR MASK
01DCD E4410 1F607 18447            CTL HIO,HSIMGTS & JSR CHKTAG$   ;CHECK FOR AN IDR
                  18448    ;
01DCE 373B3 FFFFF 18449            RDREGB UBR2,R3                  ;READ DATA 12 MSB'S AND...
01DCF 373A4 FFFFF 18450            RDREGB UBR1,R4                  ;24 LSB'S
01DD0 E457F FFFFF 18451            CTL HIO,SIDACK                  ;ACKNOWLEDGE THIS WORD AND...
01DD1 133F2 FFFFF 18452            DECRB    R2                     ;Decr. word count
01DD2 E45FF 1F611 18453            CTL HIO,CLRHSIM & JSR RDNEXT$   ;GET THE REFERENCE DATA
01DD3 61140 FFFFF 18454            XOR AB,FOUTL,R4,R0              ;DATA 24 LSB'S O.K?
01DD4 60130 33DD6 18455            XOR AQ,FOUTL,R3,ZERO & BNZ $+2  ;NO, JUMP; 12 MSB'S O.K?
01DD5 333F2 335D8 18456            OR       ZB,B,,R2 & BOZ $+3     ;Yes, jump; Check word count
01DD6 0B3F8 FFFFF 18457            INCRB    R8                     ;Incr. data error count
```

HSSM DATA PATH CONFIDENCE TEST

```
 Addr             Line
01DD7 333F2 FFFFF 18458              OR       ZB,B,,R2                 ;Check word count
01DD8 34940 33DCC 18459              ALDREGU R4,ZERO & BNZ PATHCHK2$    ;IF NON-ZERO, READ NEXT WORD
                  18460    ;
01DD9 E4420 FFFFF 18461              CTL HIO,HSIMGTR                  ;CLEAR HAIO SIMULATE TAGS
01DDA 3705F F000F 18462              RDMSKQ H#F
01DDB 46363 FFFFF 18463              AND DQ,B,SMCST,R3                ;ANY PARITY ERROR?
                  18464              BNZ.P IRPT12+1                   ;YES, PROCESS IT
01DDC E0400 33BBA 18464 +            SETPGE IRPT12+1_SHR_H#B & BNZP IRPT12+1
                  18465    ;
01DDD 37352 F1DDF 18466    PATHCHK3: RDMSKB R2,$+D#2
                  18467              JSR.P UIRPT                      ;CHECK FOR U-COMPLETE
01DDE E0400 1F372 18467 +            SETPGE UIRPT_SHR_H#B & JSRP UIRPT
                  18468              JSR.P UTIMER                     ;IT'S NOT, DECR. TIMER...
01DDF E0400 1F4E6 18468 +            SETPGE UTIMER_SHR_H#B & JSRP UTIMER
01DE0 FFFFF E1DDD 18469              JMP PATHCHK3                     ;AND CHECK AGAIN
                  18470    ;
01DE1 333F8 FFFFF 18471              OR       ZB,B,,R8                ;Any data errors?
01DE2 C10C0 33E25 18472              RD HSSM,RDDREC,R0 & BNZ PATHERR4  ;Yes, error out
01DE3 C10C0 FFFFF 18473              RD HSSM,RDDREC,R0                ;Read the test status
01DE4 FFFFF 33E14 18474              BNZ PATHERR1$                    ;IF NON-ZERO, PROCESS ERROR
                  18475              JSR.P CLRHSSMA                   ;CLEAR THE HSSM BOARDS
01DE5 E0400 1F342 18475 +            SETPGE CLRHSSMA_SHR_H#B & JSRP CLRHSSMA
```

Addr Line HSSM DATA PATH CONFIDENCE TEST

```
                18477   ;    THE DRIVE POSITION HAS BEEN VERIFIED.  THE DATA PATH FROM THE DIF, THRU
                18478   ;THE NDB AND THE XMC1 BYTE-TO-WORD CONVERTER HAS BEEN VERIFIED.  NOW THE DATA
                18479   ;PATH IN THE OPPOSITE DIRECTION WILL BE CHECKED.
                18480   ;
01DE6 B1F00 F002D 18481           WRTE HSSM,LDXMCJ & NOP WD2BYTE
01DE7 E100D FFFFF 18482           CTL HSSM,XMCJCMD                ;INITIATE WORD-TO-BYTE TEST
01DE8 3715E F03EF 18483           MOVE MSK2,IREG & NOP WORD0-1    ;LOC. OF REFERENCE WORD 0
01DE9 37358 F0004 18484           RDMSKB R8,4                     ;NO. OF WORDS TO WRITE
01DEA 37350 F1DFC 18485           RDMSKB R0,PATHCHK6              ;SM-DONE NORMAL RETURN
                18486           JSR.P SAVIRPT                   ;SAVE IT
01DEB E0400 1F4C0 18486 +         SETPGE SAVIRPT_SHR_H#B & JSRP SAVIRPT
01DEC 37350 F2B5E 18487           RDMSKB R0,2B5E                  ;20-MS. SM-IRPT. TIMER
                18488           JSR.P SAVSMTIM                  ;SAVE IT
01DED E0400 1F4C2 18488 +         SETPGE SAVSMTIM_SHR_H#B & JSRP SAVSMTIM
01DEE E4410 FFFFF 18489           CTL HIO,HSIMGTS                 ;ENABLE HAIO SIMULATE TAGS
01DEF 37354 F0002 18490           RDMSKB R4,2                     ;SET THE ODREQ MASK BIT
                18491   ;
01DF0 E4402 1F611 18492   PATHCHK5: CTL HIO,HDRVGTR & JSR RDNEXT$  ;GET THE NEXT DATA WORD
01DF1 721FC FFFFF 18493           NQLDREG UBR3                    ;LOAD U-BUS DATA BITS 35-24
01DF2 7490B FFFFF 18494           NALDREGU R0,UBR2                ;LOAD U-BUS DATA BITS 23-12
01DF3 7410A 1F607 18495           NALDREG R0,UBR1 & JSR CHKTAG$   ;LOAD BITS 11-0 & CHECK FOR ODREQ
                18496   ;
01DF4 E4401 FFFFF 18497           CTL HIO,HDRVGTS                 ;ENABLE DATA TO THE U-BUS
01DF5 E45BF FFFFF 18498           CTL HIO,SODACK                  ;ACKNOWLEDGE THE WORD
01DF6 133F8 FFFFF 18499           DECRB R8                        ;DECR. WORD COUNT
01DF7 E45FF 33DF0 18500           CTL HIO,CLRHSIM & BNZ PATHCHK5$ ;IF NOT 0, LOAD NEXT WORD
                18501   ;
01DF8 E4402 FFFFF 18502           CTL HIO,HDRVGTR                 ;DISABLE DATA FROM U-BUS
                18503           JSR.P SMIRPT                    ;CHECK FOR SM-DONE
01DF9 E0400 1F37C 18503 +         SETPGE SMIRPT_SHR_H#B & JSRP SMIRPT
                18504           JSR.P SMTIMER                   ;IT'S NOT, DECR. TIMER...
01DFA E0400 1F4DB 18504 +         SETPGE SMTIMER_SHR_H#B & JSRP SMTIMER
01DFB FFFFF E1DF9 18505           JMP $-2                         ;AND CHECK AGAIN
                18506   ;
01DFC E4420 4F001 18507   PATHCHK6: CTL HIO,HSIMGTR & PSHLD 1      ;DISABLE HAIO SIMULATE TAGS
01DFD C10C0 887FF 18508           RD HSSM,RDDREC,R0 & CLPCNT EVCNTZ ;SM-DONE; READ THE TEST STATUS
01DFE 333FA 33E14 18509           OR ZB,B,,R10 & BNZ PATHERR1$    ;IF NON-ZERO, PROCESS THE ERROR
01DFF 0F30A 33E01 18510           ADD DZC,B,ZERO,R10 & BNZ PATHCK6A ;IF SECOND TEST, DONE; SET 2ND TEST
01E00 343B4 E1DB9 18518           RTOR R11,R4 & JMP PATHCHK0      ;DO THE SECOND TEST
                18520   ;
                18521
                18522   ;THE HSSM DATA CONFIDENCE DIAGNOSTIC HAS COMPLETED SUCCESSFULLY.
                18523   ;
01E01 B1F00 F0002 18524   PATHCK6A: WRTE HSSM,LDXMCJ & NOP HSWAIT   ;HSSM WAIT LOOP
01E02 E100D FFFFF 18525           CTL HSSM,XMCJCMD                ;DO IT
01E03 E46F1 FFFFF 18526   PATHCHK7: CTL HIO,ENCHT                  ;ENABLE CHANNEL TAGS
01E04 3710F F0009 18527           RESETRAM SEN109                 ;RESET SENSE 9, FORMAT 1 AND...
```

HSSM DATA PATH CONFIDENCE TEST

```
 Addr            Line
01E05 3710F F0021 18528            RESETRAM SEN209                    ;SENSE 9, FORMAT 2
01E06 3437A AF7FF 18530            RTOR     R7,R10 & RTS              ;$Return
                  18540
01E07 3705F F0130 18541   CHKTAG:  RDMSKQ 130                         ;STANDARD SM-SIDE MASK
01E08 46110 C0064 18542            AND DQ,FOUTL,INTR1,ZERO & LDCT 100 ;ANY ALLOWED INTERRUPT?
01E09 FFFFF 33E0D 18543            BNZ CHKTAG2$                       ;YES, PROCESS IT
01E0A 451C4 B8623 18544   CHKTAG1: AND DB,FOUTL,HSIMST,R4 & CJPOP EVCNTZ,PATHERR3$ ;EXPECTED TAG?
01E0B 46110 93E09 18545            AND DQ,FOUTL,INTR1,ZERO & CJPCNT NALUZ,$-2 ;NO, CHECK AGAIN
01E0C FFFFF AF7FF 18546            RTS                                ;EXPECTED TAG RECEIVED, RETURN
                  18547   ;
01E0D E4402 FFFFF 18548   CHKTAG2: CTL H10,HDRVGTR                    ;DISABLE DATA FROM U-BUS
                  18549            JSR.P SM1RPT+1                     ;PROCESS AN ERROR
01E0E E0400 1F37D 18549 +          SETPGE SM1RPT+1_SHR_H#B & JSRP SM1RPT+1
01E0F 3705F F0130 18550            RDMSKQ 130                         ;ERROR COULD NOT BE PROCESSED,
01E10 FFFFF E1E0A 18551            JMP CHKTAG1                        ;CONTINUE CHECKING
                  18552   ;
01E11 0F1EE E003B 18553   RDNEXT: INCRD IREG,IREG & JMP SMBRD
```

Addr Line HSSM DATA PATH CONFIDENCE TEST

```
                    18555
                    18556   ;  ----- CONFIDENCE TEST ERROR PROCESSING -----
                    18557   ;
                    18558   ;    THIS CODE SETS UP THE 3 LSB'S OF SENSE BYTE 9, FORMAT 2 TO DEFINE THE
                    18559   ;FAILURE. THE FAILURE CODE VALUE OF THESE BITS MEANS:
                    18560   ;    0 - SEE OTHER BYTES
                    18561   ;    1 - WORD-TO-BYTE MISCOMPARE
                    18562   ;    2 - BYTE-TO-WORD MISCOMPARE
                    18563   ;    3 - DMA FAILURE (EARLY COMPLETE)
                    18564   ;    4 - U-SIDE DMA FAILURE (NO COMPLETE)
                    18565   ;    5 - HEADER FLAG MISCOMPARE
                    18566   ;    6 - ODR OR IDR TAG MISSING
                    18567   ;    7 - UNDEFINED ERROR
                    18568
01E12 C10C0 FFFFF 18569   PATHERR0: RD HSSM,RDDREC,R0                ;SM-DONE (ERROR) IRPT.
01E13 C10C0 8F7FF 18570             RD HSSM,RDDREC,R0 & POP          ;READ ERROR STATUS
                    18571   ;
01E14 FFFFF 5F629 18572   PATHERR1: LDAR PATHERR7$
01E15 45050 F0020 18573             TSTRBIT R0,20                    ;EARLY COMPLETE?
01E16 FFFFF 33618 18574             BOZ $+2                          ;NO, JUMP
01E17 37341 5F803 18575             RDREGB MSK1,R1 & JAR H#3         ;FAILURE CODE=3
01E18 45050 F0040 18576             TSTRBIT R0,40                    ;NO COMPLETION?
01E19 FFFFF 3361B 18577             BOZ $+2                          ;NO, JUMP
01E1A 37341 5F804 18578             RDREGB MSK1,R1 & JAR H#4         ;FAILURE CODE=4
01E1B 45050 F0000 18579             TSTRBIT R0,00                    ;(spare)
01E1C FFFFF 3361E 18580             BOZ $+2                          ;NO, JUMP
01E1D 37341 5F800 18581             RDREGB MSK1,R1 & JAR H#0         ;FAILURE CODE=
01E1E 45050 F0001 18582             TSTRBIT R0,1                     ;DATA MISCOMPARE?
01E1F FFFFF 33627 18583             BOZ PATHERR5$                    ;IF NOT, THIS IS IMPOSSIBLE
01E20 37341 5F801 18584             RDREGB MSK1,R1 & JAR 1           ;FAILURE CODE=1
                    18585   ;
01E21 37351 F0005 18586   PATHERR2: RDMSKB R1,5                      ;HEADER FLAG MISCOMPARE
01E22 FFFFF E1E29 18587             JMP PATHERR7
                    18588   ;
01E23 37351 F0006 18589   PATHERR3: RDMSKB R1,6                      ;ODR OR IDR TAG MISSING
01E24 E4402 E1E29 18590             CTL HIO,HDRVGTR & JMP PATHERR7
                    18591   ;
01E25 37351 F0002 18592   PATHERR4: RDMSKB R1,2                      ;BYTE-TO-WORD MISCOMPARE
01E26 FFFFF E1E29 18593             JMP PATHERR7
                    18594   ;
01E27 37351 F0007 18595   PATHERR5: RDMSKB R1,7                      ;UNDEFINED ERROR
01E28 FFFFF E1E2A 18596             JMP PATHERR7+1
                    18597   ;
01E29 3710F F0018 18598   PATHERR7: RESETRAM SEN415                  ;RESET SM-RETRY COUNT
01E2A 370FF F0021 18599             RDRAMQ SEN209                    ;GET SENSE 9, FORMAT 2 AND...
01E2B 3011F F0021 18600             OR AQ,FOUTL,R1,RAME & NOP SEN209 ;INCLUDE THE FAILURE CODE
                    18601   PATHERR8: JSR.P CLRHSSM                  ;CLEAR HSSM BOARDS
01E2C E0400 1F340 18601 +           SETPGE CLRHSSM_SHR_H#B & JSRP CLRHSSM
01E2D E4420 FFFFF 18602             CTL HIO,HSIMGTR                  ;DISABLE HAIU SIMULATE TAGS
01E2E E46F1 E04AD 18603             CTL HIO,ENCHT & JMP READ62       ;ENABLE CHANNEL TAGS AND...
                    18604   ;                                         SEND PERMANENT EQUIP. CHECK
```

Addr Line HSSM DATA PATH CONFIDENCE TEST

```
                    18606
                    18607    ;    THIS ROUTINE ASSEMBLES 4 REFERENCE WORDS AND STORES THEM IN SMB MEMORY
                    18608    ;(DISPLAYS 550-553) FOR USE LATER.
                    18609    ;
                    18610    ;UPON ENTRY:  R8  = HEADER FLAGS
                    18611    ;             R4  = CYLINDER
                    18612    ;             R12 = HEAD
                    18613    ;
                    18614
01E2F 0F1EE E0044   18615    WRTNEXT:  INCRD IREG,IREG & JMP SMBWRT
                    18616    ;
01E30 433F0 FFFFF   18627    ASMWRDS:  RESETR R0                         ;NO DRIVE NO. IN HEADER OPTION
01E31 45050 F0060   18629              TSTRBIT R0,60                      ;ISOLATE THE MSB'S IN Q
01E32 303C1 FFFFF   18630              OR AQ,B,R12,R1                     ;MERGE WITH HEAD INTO R1
01E33 60700 4F004   18631              XOR AQ,LSBZ,R0,R0 & PSHLD 4        ;DRIVE LSB'S TO R0 AND...
                    18632              LSHFT2 R0,R0                       ;MOVE THEM INTO POSITION
01E34 01700 887FF   18632 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01E35 31040 FFFFF   18633              OR AB,Q,R4,R0                      ;MERGE WITH CYLINDER INTO Q
01E36 4614E F00FF   18634              AND DQ,FOUTL,MSK1,IREG & NOP H#FF  ;CYL. 8 LSB'S TO IREG AND...
01E37 660EF FFFFF   18635              XOR DQ,Q,IREG                      ;8 MSB'S TO Q; STROBE XFER. REG.
01E38 D0FF0 4F001   18636              RDXFR SMC,,R0 & PSHLD 1            ;8 LSB'S TO R0-UPPER AND...
                    18637              LSHFT2 R0,R0                       ;MOVE THEN INTO POSITION
01E39 01700 887FF   18637 +            ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01E3A 30300 C0003   18638              OR AQ,B,R0,R0 & LDCT 3             ;MERGE THE MSB'S
01E3B 45058 F00FF   18639              TSTRBIT R8,H#FF                    ;ISOLATE DRIVE FLAGS IN Q
01E3C 336F0 9863C   18640              OR ZB,LSBQZ,,R0 & CJPCNT EVCNTZ,$  ;POSITION FLAGS & CYLINDER
                    18641    ;
01E3D 01711 FFFFF   18642              ADD AB,LSBZ,R1,R1                  ;MOVE THE HEAD...
01E3E 01711 FFFFF   18643              ADD AB,LSBZ,R1,R1                  ;NUMBER INTO POSITION
01E3F 31310 FFFFF   18644              OR AB,B,R1,R0                      ;MERGE WITH CYLINDER
01E40 3715E F03F0   18645              MOVE MSK2,IREG & NOP WORD0         ;LOC. OF FIRST WORD
01E41 45150 F0010   18646              TSTBIT R0,10                       ;ODD HEAD NUMBER?
01E42 FFFFF 33E51   18647              BNZ ASMODD$                        ;YES, JUMP
                    18648    ;
01E43 35350 F0005   18649              SETRBIT R0,5                       ;INCLUDE EVEN LSB'S (NIBBLE)
01E44 FFFFF 1F37D   18650              JSR SMBWRTD                        ;SAVE WORD 0
                    18651    ;
01E45 3705F F0555   18652              RDMSKQ 555                         ;WORD 1 MSB'S = 555
01E46 D0FF0 FFFFF   18653              RDXFR SMC,,R0                      ;MIDDLE BITS = 555
01E47 35350 F0555   18654              SETRBIT R0,555                     ;LOWER BITS = 555
01E48 FFFFF 1F62F   18655              JSR WRTNEXT$                       ;SAVE WORD 1
                    18656    ;
01E49 3705F F0222   18657              RDMSKQ 222                         ;WORD 2 MSB'S = 222
01E4A D0FF0 FFFFF   18658              RDXFR SMC,,R0                      ;MIDDLE BITS = 222
01E4B 35350 F022A   18659              SETRBIT R0,22A                     ;LOWER BITS = 22A
01E4C FFFFF 1F62F   18660              JSR WRTNEXT$                       ;SAVE WORD 2
                    18661    ;
01E4D 3705F F0AAA   18662              RDMSKQ H#AAA                       ;WORD 3 MSB'S = AAA
01E4E D0FF0 FFFFF   18663              RDXFR SMC,,R0                      ;MIDDLE BITS = AAA
```

Addr Line HSSM DATA PATH CONFIDENCE TEST

```
01E4F 35350 F0AAA 18664            SETRBIT R0,H#AAA                ;LOWER BITS = AAA
01E50 FFFFF E1E2F 18665            JMP WRTNEXT                     ;SAVE WORD 3 & RETURN
                  18666    ;
01E51 35350 F0002 18667    ASMODD: SETRBIT R0,2                    ;INCLUDE ODD LSB'S (NIBBLE)
01E52 FFFFF 1F37D 18668            JSR SMBWRTD                     ;SAVE WORD 0
                  18669    ;
01E53 3705F F0222 18670            RDMSKQ 222                      ;WORD 1 MSB'S = 222
01E54 D0FF0 FFFFF 18671            RDXFR SMC,,R0                   ;MIDDLE BITS = 222
01E55 35350 F02AA 18672            SETRBIT R0,2AA                  ;LOWER BITS = 2AA
01E56 FFFFF 1F62F 18673            JSR WRTNEXT$                    ;SAVE WORD 1
                  18674    ;
01E57 3705F F0AAA 18675            RDMSKQ H#AAA                    ;WORD 2 MSB'S = AAA
01E58 D0FF0 FFFFF 18676            RDXFR SMC,,R0                   ;MIDDLE BITS = AAA
01E59 35350 F0AA5 18677            SETRBIT R0,H#AA5                ;LOWER BITS = AA5
01E5A FFFFF 1F62F 18678            JSR WRTNEXT$                    ;SAVE WORD 2
                  18679    ;
01E5B 3705F F0555 18680            RDMSKQ 555                      ;WORD 3 MSB'S = 555
01E5C D0FF0 FFFFF 18681            RDXFR SMC,,R0                   ;MIDDLE BITS = 555
01E5D 35350 F0555 18682            SETRBIT R0,555                  ;LOWER BITS = 555
01E5E FFFFF E1E2F 18683            JMP WRTNEXT                     ;SAVE WORD 3 & RETURN
                  18688
                  18689
                  18690    TITLE MORE MISCELLANEOUS ROUTINES
```

Addr Line MORE MISCELLANEOUS ROUTINES

```
                    18710   ;┌──────────────────────────────────────────────────────
                    18711   ;| Display one word of MDB data on the diagnostic display panel
                    18712   ;|
                    18713   ;| INPUT PARAMETERS:
                    18714   ;|      R0: Contains the address of the location to be displayed.
                    18715   ;|          (NOTE: addresses 000 to 3FF are valid)
                    18716   ;|
                    18717   ;| REGISTERS USED:  R0,R1,R2 & Q.
                    18718   ;└──────────────────────────────────────────────────────
01E5F 37160 1F664   18719   SMDISPA:  MOVE SMCST,ZERO & JSR MDBRD       ;READ ONE WORD FROM MDB
01E60 323F1 FFFFF   18720             QTOR R1                           ;MOVE 12 MSB'S TO R1
01E61 9351F FFFFF   18721             WRTL SCP,LDDISP1,R1               ;DISPLAY BITS 24-35
01E62 A360F FFFFF   18722             WRTU SCP,LDDISP2,R0               ;DISPLAY BITS 12-23
01E63 9370F 5FFFF   18723             WRTL SCP,LDDISP3,R0 & JAR         ;DISPLAY BITS 0-11 & RETURN
                    18724
                    18725   TITLE READ ONE WORD FROM MDB
```

Addr Line READ ONE WORD FROM MDB

```
                18727   ;┌──────────────────────────────────────────────────
                18728   ;|   READ ONE WORD FROM MDB MEMORY.
                18729   ;|
                18730   ;|UPON ENTRY:  R0 = WORD NO. TO BE READ
                18731   ;|UPON EXIT:   Q  = DATA BITS 35-24
                18732   ;|            R0 = DATA BITS 23- 0
                18733   ;|REGISTERS USED:  R0,R1,R2 & Q.
                18734   ;└──────────────────────────────────────────────────
01E64 45050 F0001 18735   MDBRD:    TSTRBIT R0,1                    ;ODD WORD NOUNBER?
01E65 323F2 FFFFF 18736            QTOR R2                         ;SAVE ODD WORD STATUS IN R2
                18737              JSR.P FOUR5DN                   ;COMPUTE BYTE ADDRESS
01E66 E0400 1F368 18737 +          SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
01E67 A211F FFFFF 18738            WRTU HSSMA,LDMDARM,R1           ;LOAD THE MSB'S
01E68 45340 F00FF 18739            AND DB,B,MSK1,R0 & NOP H#FF     ;ISOLATE THE LSB'S...
01E69 9200F 4F001 18740            WRTL HSSMA,LDMDARL,R0 & PSHLD 1 ;AND LOAD THEM
                18741   ;
01E6A C10D0 887FF 18742            RD HSSM,RDMDBDA,R0 & CLPCNT EVCNTZ ;READ THE 8 MSB'S AND...
01E6B D10D0 FFFFF 18743            RDXFR HSSM,RDMDBDA,R0           ;MOVE THEM TO R0-UPPER
01E6C C10D1 4F001 18744            RD HSSM,RDMDBDA,R1 & PSHLD 1    ;READ THE 2ND 8 BITS AND...
                18745              LSHFT2 R1,R1                    ;LEFT SHIFT THEM 4 PLACES
01E6D 01711 887FF 18745 +          ADD AB,LSBZ,R1,R1 & CLPCNT EVCNTZ
01E6E 55350 F00FF 18746            RSTRBIT R0,H#FF                 ;DELETE POSSIBLE BAD DATA
01E6F 31310 4F001 18747            OR AB,B,R1,R0 & PSHLD 1         ;MERGE THE 2ND 8 BITS
                18748              LSHFT2 R0,R0                    ;LEFT SHIFT 16 BITS, 4 PLACES
01E70 01700 887FF 18748 +          ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01E71 C10D1 FFFFF 18749            RD HSSM,RDMDBDA,R1
01E72 C10D1 FFFFF 18750            RD HSSM,RDMDBDA,R1              ;READ THE 3RD 8 BITS,
01E73 31310 4F007 18751            OR AB,B,R1,R0 & PSHLD 7         ;MERGE THEM WITH THE 1ST 16 AND...
01E74 336F0 887FF 18752            LSHFTDZ R0                      ;LEFT SHIFT 8 PLACES
                18753   ;
01E75 C10D1 FFFFF 18754            RD HSSM,RDMDBDA,R1
01E76 C10D1 FFFFF 18755            RD HSSM,RDMDBDA,R1              ;GET THE 4TH 8 BITS,
01E77 31310 4F003 18756            OR AB,B,R1,R0 & PSHLD 3         ;MERGE THEN WITH THE 1ST 24 AND...
01E78 336F0 887FF 18757            LSHFTDZ R0                      ;LEFT SHIFT 4 PLACES
                18758   ;
01E79 C10D1 FFFFF 18759            RD HSSM,RDMDBDA,R1
01E7A C10D1 FFFFF 18760            RD HSSM,RDMDBDA,R1              ;READ THE LSB'S
01E7B 333F2 C0003 18761            OR ZB,B,,R2 & LDCT 3            ;ODD WORD NUMBER?
01E7C 37160 33E7F 18762            MOVE SMCST,ZERO & BNZ $+3       ;YES, JUMP; reset XMC parity
01E7D 335F1 9867D 18763            OR ZB,RSBZ,,R1 & CJPCNT EVCNTZ,$ ;RIGHT SHIFT LSB'S 4 PLACES
01E7E 31310 AF7FF 18764            OR AB,B,R1,R0 & RTS             ;MERGE WITH 32 MSB'S & RETURN
                18765   ;
01E7F 336F0 9867F 18766            OR ZB,LSBQZ,,R0 & CJPCNT EVCNTZ,$ ;SHIFT 32 BITS 4 MORE PLACES,
01E80 31310 AF7FF 18767            OR AB,B,R1,R0 & RTS             ;MERGE THE 8 LSB'S & RETURN
                18768
                18769   TITLE
```

```
 Addr             Line


                  18771
                  18772    READDONE: EQU $
01E81 37351 F2710 18777              RDMSKB R1,2710                    ;SET 10-MS. TIMER
01E82 3705F F0285 18778              RDMSKQ 285                        ;SET INTERRUPT MASK
01E83 46383 5F683 18779              AND DQ,B,HIOST,R3 & LDAR $        ;ANY ALLOWED BITS?
                  18780              BOZ.P READ22                      ;NO, WE'RE DONE
01E84 E0400 33480 18780 +            SETPGE READ22_SHR_H#B & BOZP READ22
01E85 65153 F0080 18781              TSTREQU R3,H#80                   ;DATA IN BUFFER ONLY?
01E86 133F1 33E88 18782              DECRB R1 & BNZ $+2                ;NO, CHECK FURTHER; DECR. TIMER
                  18783              CJARB.P ALUZ,READ22               ;IF NOT TIME-OUT, CHECK AGAIN
01E87 E0400 73480 18783 +            SETPGE READ22_SHR_H#B & CJARBP ALUZ,READ22
                  18784    ;
01E88 37352 F047E 18785              RDMSKB R2,READ22-D#2              ;ERROR RETURN LOC.
01E89 45053 F0005 18786              TSTRBIT R3,5                      ;PARITY OR E.F?
                  18787              BNZ.P IRPT11                      ;YES, PROCESS IT
01E8A E0400 33BAE 18787 +            SETPGE IRPT11_SHR_H#B & BNZP IRPT11
01E8B FFFFF E03FF 18788              JMP IRPT31                        ;PROCESS U-RESPONSE TIME-OUT
                  18789
                  18833    ;
                  18834    ;    THIS ROUTINE WILL PRINT 3 HEX CHARACTERS.
                  18835    ;IT IS ENTERED WITH:  CALL R2,PRINT3H
                  18836    ;UPON ENTRY:  R0 = CHARS. TO BE PRINTED
                  18837    ;
01E8C 37351 F0003 18838    PRINT3H:  RDMSKB R1,3                       ;3 CHARACTERS
01E8D 3705F F0008 18839              RDMSKQ 8                          ;PRINT HEX NUMBER
01E8E FFFFF E0E7C 18840              JMP ISSCMDA                       ;DO IT
                  18841
                  18842    TITLE WRITE ONE WORD TO MDB
```

Addr Line WRITE ONE WORD TO MDB

```
                18844   ;
                18845   ;UPON ENTRY:  R0 = WORD NUMBER TO BE WRITTEN
                18846   ;             R3 = DATA 24 LSB'S
                18847   ;             Q  = DATA 12 MSB'S
                18848   ;REGISTERS USED:  R0,R1,R2 & Q.
                18849   ;
01EBF 34302 FFFFF 18850   MDBWRT:  RTOR R0,R2                      ;TEMP. SAVE WORD NO. IN R2
                18851             JSR.P FOUR5DN                   ;COMPUTE THE BYTE ADDRESS
01E90 E0400 1F368 18851 +         SETPGE FOUR5DN_SHR_H#8 & JSRP FOUR5DN
01E91 A211F FFFFF 18852           WRTU HSSMA,LDMDARM,R1           ;LOAD THE MSB'S
01E92 45350 F00FF 18853           MSKRBIT  R0,H#FF                ;ISOLATE THE 8 LSB'S...
01E93 9200F FFFFF 18854           WRTL HSSMA,LDMDARL,R0           ;AND LOAD THEM
01E94 55351 F0FFF 18855           RSTRBIT R1,H#FFF                ;RESET SHIFTED ADDR. LSB'S
01E95 45152 F0001 18856           TSTBIT R2,1                     ;ODD WORD NUMBER?
01E96 31301 33EB5 18857           OR AB,B,R0,R1 & BNZ MDBWRTO     ;YES, JUMP; BYTE ADDR. TO R1
                18858   ;
01E97 37352 F00FF 18859           RDMSKB R2,H#FF
01E98 D0FF2 FFFFF 18860           RDXFR SMC,,R2                   ;WRITE DATA MASK (UPPER) TO R2
01E99 325F0 4F002 18861           OR Z0,RSBZ,,R0 & PSHLD 2        ;DATA 12 MSB'S TO R0
01E9A 335F0 887FF 18862           RSHFTZ R0                       ;RIGHT SHIFT 4 BITS OUT AND...
01E9B 9230F FFFFF 18863           WRTL HSSMA,LDMDAT,R0            ;WRITE THE 8 MSB'S (28-35)
                18864   ;
01E9C 34630 4F002 18865           OR AZ,LSBQZ,R3,R0 & PSHLD 2     ;24 MSB'S TO R0 AND...
01E9D 336F0 887FF 18866           LSHFTDZ R0                      ;LEFT SHIFT BITS 20-27 INTO Q
01E9E 46340 F00FF 18867           AND DQ,B,MSK1,R0 & NOP H#FF     ;ISOLATE BITS 20-27 IN R0 AND...
01E9F 9230F FFFFF 18868           WRTL HSSMA,LDMDAT,R0            ;AND WRITE THEM
                18869   ;
01EA0 41023 FFFFF 18870           AND AB,Q,R2,R3                  ;ISOLATE BITS 12-19 IN Q-UPPER
01EA1 323F0 FFFFF 18871           QTOR R0                         ;MOVE THEM TO R0 AND...
01EA2 A230F FFFFF 18872           WRTU HSSMA,LDMDAT,R0            ;WRITE BITS 12-19
                18873   ;
01EA3 34530 4F002 18874           OR AZ,RSBZ,R3,R0 & PSHLD 2      ;24 LSB'S TO R0 AND...
01EA4 335F0 887FF 18875           RSHFTZ R0                       ;RIGHT SHIFT 4 PLACES
01EA5 45350 F00FF 18876           MSKRBIT R0,H#FF                 ;ISOLATE BITS 4-11...
01EA6 9230F FFFFF 18877           WRTL HSSMA,LDMDAT,R0            ;AND WRITE THEM
                18878   ;
01EA7 45053 F000F 18879           TSTRBIT R3,H#F                  ;ISOLATE BITS 0-3 IN Q
01EA8 323F0 4F001 18880           QTOR R0 & PSHLD 1               ;MOVE TO R0 AND...
                18881             LSHFT2 R0,R0                    ;SHIFT LEFT 4 PLACES
01EA9 01700 887FF 18881 +         ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01EAA C10D2 FFFFF 18882           RD HSSM,RDMDBDA,R2              ;READ OLD DATA 4 LSB'S...
01EA8 C10D2 FFFFF 18883           RD HSSM,RDMDBDA,R2              ;AND NEXT WORD 4 MSB'S
01EAC 55352 F00F0 18884           RSTRBIT R2,H#F0                 ;DELETE OLD DATA LSB'S AND..
01EAD 31320 FFFFF 18885           OR AB,B,R2,R0                   ;REPLACE THEM WITH NEW LSB'S
01EAE 05341 F0004 18886           ADD DB,B,MSK1,R1 & NOP H#4      ;ADDRESS OF THIS BYTE
01EAF 45051 F0100 18887           TSTRBIT R1,100                  ;LSB'S OVERFLOW?
01EB0 60311 336B2 18888           XOR AQ,B,R1,R1 & BOZ $+2        ;NO, JUMP
01EB1 05351 F1000 18889           ADD DB,B,MSK2,R1 & NOP H#1000   ;INCR. ADDRESS MSB'S
01EB2 A211F FFFFF 18890           WRTU HSSMA,LDMDARM,R1
01EB3 9201F FFFFF 18891           WRTL HSSMA,LDMDARL,R1           ;LOAD THE BYTE ADDRESS AND...
01EB4 9230F AF7FF 18892           WRTL HSSMA,LDMDAT,R0 & RTS      ;WRITE BITS 0-3; RETURN
                18893
```

```
 Addr            Line WRITE ONE WORD TO MDB

01EB5 323F0 4F007 18894   MDBWRTO:  QTOR R0 & PSHLD 7                ;DATA 12 MSB'S TO R0
01EB6 335F0 887FF 18895             RSHFTZ R0                       ;SHIFT OUT ALL BUT BITS 32-35
01EB7 C10D2 FFFFF 18896             RD HSSM,RDMDBDA,R2
01EB8 C10D2 FFFFF 18897             RD HSSM,RDMDBDA,R2              ;READ THE FIRST BYTE AND...
01EB9 55352 F000F 18898             RSTRBIT R2,H#F                  ;DELETE THE OLD DATA (4 LSB'S)
01EBA A211F FFFFF 18899             WRTU HSSMA,LDMDARM,R1
01EBB 9201F FFFFF 18900             WRTL HSSMA,LDMDARL,R1           ;RELOAD THE ADDRESS
01EBC 31320 FFFFF 18901             OR AB,B,R2,R0                   ;MERGE THE NEW DATA AND...
01EBD 9230F FFFFF 18902             WRTL HSSMA,LDMDAT,R0            ;WRITE BITS 32-35
                  18903   ;
01EBE 46340 F00FF 18904             AND DQ,B,MSK1,R0 & NOP H#FF     ;ISOLATE BITS 24-31
01EBF 9230F FFFFF 18905             WRTL HSSMA,LDMDAT,R0            ;AND WRITE THEM
                  18906   ;
01EC0 34530 4F002 18907             OR AZ,RSBZ,R3,R0 & PSHLD 2      ;24 LSB'S TO R0 AND...
01EC1 335F0 887FF 18908             RSHFTZ R0                       ;RIGHT SHIFT
01EC2 37352 F00FF 18909             RDMSKB R2,H#FF
01EC3 D0FF2 FFFFF 18910             RDXFR SMC,,R2                   ;WRITE DATA MASK (UPPER) TO R2
01EC4 41320 FFFFF 18911             AND AB,B,R2,R0                  ;ISOLATE BITS 16-23...
01EC5 A230F FFFFF 18912             WRTU HSSMA,LDMDAT,R0            ;AND WRITE THEM
                  18913   ;
01EC6 34330 4F001 18914             RTOR R3,R0 & PSHLD 1            ;24 LSB'S TO R0 AND...
                  18915             LSHFT2 R0,R0                    ;LEFT SHIFT 4 PLACES
01EC7 01700 887FF 18915 +           ADD AB,LSBZ,R0,R0 & CLPCNT EVCNTZ
01EC8 41320 FFFFF 18916             AND AB,B,R2,R0                  ;ISOLATE BITS 8-15...
01EC9 A230F FFFFF 18917             WRTU HSSMA,LDMDAT,R0            ;AND WRITE THEM
                  18918   ;
01ECA 45053 F00FF 18919             TSTRBIT R3,H#FF                 ;ISOLATE BITS 0-7 IN Q,
01ECB 323F0 FFFFF 18920             QTOR R0                         ;MOVED TO R0
01ECC 9230F AF7FF 18921             WRTL HSSMA,LDMDAT,R0 & RTS      ;WRITE THEM & RETURN
                  18922
                  18923
                  18924   TITLE PHONY-UP HOME ADDRESS & COUNT
```

Addr Line PHONY-UP HOME ADDRESS & COUNT

```
                    18926
                    18927   ;     THIS ROUTINE WILL GENERATE HOME ADDRESS AND COUNT AREA DATA FROM
                    18928   ;CYLINDER AND HEAD DATA PROVIDED IN THE ADDRESS WORD.
                    18929   ;THIS ROUTINE IS USED WITH "READ FULL TRACK" AND "READ HOME ADDRESS SPECIAL".
                    18930
                    18931
01ECD 373F1 F0242 18932   READ70:   RDRAMB  R1, CASHSTAT              ;GET THE CACHING STATUS
01ECE 45051 F0001 18933             TSTRBIT R1, CACHING               ;CACHING MODE?
01ECF B2100 A3800 18934             WRTE HSSMA,LDMDARM & CRTS NALUZ,0  ;IF YES, RETURN
01ED0 B2000 F0000 18935             WRTE HSSMA,LDMDARL & NOP 0        ;CLEAR THE STARTING ADDR.
01ED1 4505E F0080 18936             TSTRBIT R14,80                    ;1-TRACK FLAG?
01ED2 FFFFF 33ED6 18937             BNZ READ71                        ;IF SET, JUMP
                    18938             JSR.P WAITCON                     ;THIS IS FOR MPC STACK ADJUSTMENT
01ED3 E0400 1F521 18938 +           SETPGE WAITCON_SHR_H#B & JSRP WAITCON
01ED4 FFFFF 4F008 18939             PSHLD 8
01ED5 B2300 88000 18940             WRTE HSSMA,LDMDAT & CLPCNT EVCNTZ,0 ;CLEAR SKIP DISPLACEMENT DATA
                    18941   ;
01ED6 B2300 F0000 18942   READ71:   WRTE HSSMA,LDMDAT & NOP 0          ;CLEAR THE FLAG BYTE
01ED7 373F4 F0006 18943             RDRAMB R4,SEN06                   ;GET CYL. MSB' & HEAD INTO R4
01ED8 45054 F001F 18944             TSTRBIT R4,1F                     ;ISOLATE HEAD NO. IN Q AND...
01ED9 60542 4F003 18945             XOR AQ,RSBZ,R4,R2 & PSHLD 3       ;CYL. MSB'S IN R2
01EDA 335F2 887FF 18946             RSHFTZ R2                         ;MOVE CYL. MSB'S INTO POSITION
01EDB 323F4 FFFFF 18947             QTOR R4                           ;HEAD ONLY TO R4
                    18948   ;
01EDC 373F3 F0005 18950             RDRAMB R3,SEN05                   ;<CYLINDER LSB'S TO R3
01EDD 9232F FFFFF 18954             WRTL HSSMA,LDMDAT,R2              ;WRITE H.A. CYL. MSB'S
01EDE 9233F 1F6FF 18955             WRTL HSSMA,LDMDAT,R3 & JSR WRT0BYT$ ;WRITE H.A. CYL. LSB'S
01EDF 9234F 4F003 18956             WRTL HSSMA,LDMDAT,R4 & PSHLD 3    ;WRITE H.A. HEAD LSB'S
01EE0 B2300 88000 18957             WRTE HSSMA,LDMDAT & CLPCNT EVCNTZ,0 ;WRITE 4 ZEROS BYTES
01EE1 E1006 FFFFF 18958             CTL HSSM,RSMDBPER                 ;RESET SMB PARITY ERRORS
01EE2 34378 FFFFF 18959             RTOR R7,R8                        ;U-XFER LENGTH TO R8
01EE3 4505E F0080 18960             TSTRBIT R14,80                    ;1-TRACK FLAG?
                    18961             CJPOP.P ALUZ,READ10               ;IF NOT, SEND H.A. TO CHANNEL
01EE4 E0400 B343F 18961 +           SETPGE READ10_SHR_H#B & CJPOPP ALUZ,READ10
                    18962   ;
                    18963   ;    PHONY-UP COUNT AREA DATA FOR ONE TRACK.
                    18964   ;
01EE5 37359 F0001 18965             RDMSKB R9,1                       ;FIRST RECORD IS 1
01EE6 373F0 F02CD 18966             RDRAMB R0,RECLNTH                 ;GET RECORD LENGTH AND...
                    18967             JSR.P FOUR5DN                     ;CONVERT IT TO BYTE LENGTH
01EE7 E0400 1F368 18967 +           SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
01EE8 37358 F0009 18968             RDMSKB R8,9                       ;COUNT AREA DATA STARTS AT BYTE 9
01EE9 B1000 F0012 18969             WRTE HSSM,LDSARL & NOP D#18       ;DISK DATA STARTS AT BYTE 18
01EEA 3705F F00FF 18970             RDMSKQ H#FF                       ;SET 8-BIT MASK IN Q
                    18971   ;
01EEB 34385 4F001 18972   READ72:   RTOR R8,R5 & PSHLD 1
                    18973             LSHFT2 R5,R5                      ;MOVE MSB'S INTO POSITION AND...
01EEC 01755 887FF 18973 +           ADD AB,LSBZ,R5,R5 & CLPCNT EVCNTZ
```

Addr Line PHONY-UP HOME ADDRESS & COUNT

```
01EED A215F FFFFF 18974            WRTU HSSMA,LDMDARM,R5              ;LOAD THEM
01EEE 40385 FFFFF 18975            AND AQ,B,RB,R5                    ;ISOLATE ADDR. B LSB'S...
01EEF 9205F FFFFF 18976            WRTL HSSMA,LDMDARL,R5             ;AND LOAD THEM
01EF0 9232F FFFFF 18977            WRTL HSSMA,LDMDAT,R2              ;WRITE COUNT CYL. MSB'S
01EF1 9233F 1F6FF 18978            WRTL HSSMA,LDMDAT,R3 & JSR WRT0BYT$ ;WRITE COUNT CYL. LSB'S
01EF2 9234F FFFFF 18979            WRTL HSSMA,LDMDAT,R4              ;WRITE COUNT HEAD LSB'S
01EF3 9239F 1F6FF 18980            WRTL HSSMA,LDMDAT,R9 & JSR WRT0BYT$ ;WRITE RECORD NO.
01EF4 A231F FFFFF 18981            WRTU HSSMA,LDMDAT,R1              ;WRITE DATA LENGTH MSB'S
01EF5 40306 FFFFF 18982            AND AQ,B,R0,R6
01EF6 9236F 1F6FF 18983            WRTL HSSMA,LDMDAT,R6 & JSR WRT0BYT$ ;WRITE DATA LENGTH LSB'S
                  18984    ;
01EF7 2D1F9 F02D6 18986            SUBB DB,FOUTL,RAME,R9 & NOP AMPREC ;<ARE WE AT MAX. RECORD?
01EF8 0B3F9 34EFB 18990            INCR8 R9 & CJMP NAGB,READ73       ;IF YES, JUMP; INCR. RECORD NO.
01EF9 05348 F0009 18991            ADD DB,B,MSK1,R8 & NOP H#9        ;LOC. OF DATA AREA...
01EFA 01308 E1EEB 18992            ADD AB,B,R0,R8 & JMP READ72       ;+ BYTE LENGTH = NEXT COUNT LOC.
                  18993    ;
01EFB 37359 F0001 18994    READ73: RDMSKB R9,1                       ;FIRST RECORD = 1
01EFC E1006 FFFFF 18995            CTL HSSM,RSMDBPER                 ;RESET MDB PARITY ERRORS
01EFD 433F6 FFFFF 18996            RESETR R6                         ;RESET START WORD
01EFE 433F5 AF7FF 18997            RESETR R5 & RTS                   ;RESET READ/WRITE FLAGS & RETURN
                  18998    ;
01EFF B2300 AF000 18999    WRT0BYT: WRTE HSSMA,LDMDAT & RTS 0         ;WRITE A 0 BYTE
                  19000
                  19001
                  19059    TITLE CONVERT DISK HEADER TO COUNT
```

Addr Line CONVERT DISK HEADER TO COUNT

```
                19061
01F00 5535E F0010 19062    RDCOUNT:  RSTRBIT R14,10                    ;RESET READ COUNT FLAG
01F01 B2100 F0000 19063              WRTE HSSMA,LDMDARM & NOP 0         ;SET THE BUFFER ADDRESS...
01F02 B2000 F0000 19064              WRTE HSSMA,LDMDARL & NOP 0         ;FOR THE COUNT INFO
                19065                JSR.P READCYL                      ;READ THE CURRENT CYLINDER...
01F03 E0402 1F579 19065 +            SETPGE READCYL_SHR_H#B & JSRP READCYL
01F04 A231F 1F715 19066              WRTU HSSMA,LDMDAT,R1 & JSR RDCOUNTC$ ;AND LOAD IT
01F05 FFFFF 1F6FF 19067              JSR WRT0BYT$                       ;HEAD MSB'S ARE ALWAYS 0
01F06 C10B0 FFFFF 19068              RD HSSM,RDDRHD,R0
01F07 C10D0 1F714 19069              RD HSSM,RDDRHD,R0 & JSR RDCOUNTB$  ;READ AND LOAD THE HEAD
01F08 C10C0 FFFFF 19070              RD HSSM,RDDREC,R0
01F09 C10C0 1F715 19071              RD HSSM,RDDREC,R0 & JSR RDCOUNTC$  ;READ AND LOAD THE RECORD NO.
01F0A FFFFF 1F6FF 19072              JSR WRT0BYT$                       ;THE KEY LENGTH = 0
                19073    ;
01F0B 373F0 F02CD 19074              RDRAMB R0,RECLNTH                  ;GET THE RECORD LENGTH AND...
                19075                JSR.P FOUR5DN                      ;CONVERT WORDS TO BYTES
01F0C E0400 1F368 19075 +            SETPGE FOUR5DN_SHR_H#B & JSRP FOUR5DN
01F0D A231F 1F715 19076              WRTU HSSMA,LDMDAT,R1 & JSR RDCOUNTC$ ;LOAD DATA LENGTH MSB'S & LSB'S
01F0E 433F6 1F6FF 19077              RESETR R6 & JSR WRT0BYT$           ;LAST BYTE IS 0
                19078                JSR.P NXTUADDR                     ;U-SIDE DATA STARTS AT 0000
01F0F E0400 1F4B0 19078 +            SETPGE NXTUADDR_SHR_H#B & JSRP NXTUADDR
                19079    ;
01F10 B1100 F000F 19080    RDCOUNT1: WRTE HSSM,LDSARM & NOP H#F         ;LOAD SM-ADDRESS MSB'S = 0F
01F11 B1F00 F0065 19081              WRTE HSSM,LDXMCJ & NOP H#65        ;MOVE SM-ADDRESS TO THE MDB
                19082                JSR.P HSWAITR1                     ;DO IT & DELAY
01F12 E0400 1F52D 19082 +            SETPGE HSWAITR1_SHR_H#B & JSRP HSWAITR1
01F13 FFFFF E043F 19083              JMP READ10                         ;NOW SEND COUNT TO THE CHANNEL
                19084    ;
01F14 45350 F001F 19085    RDCOUNTB: MSKRBIT R0,1F                      ;ISOLATE HEAD NO. ONLY
01F15 45350 F00FF 19086    RDCOUNTC: MSKRBIT R0,H#FF                    ;ISOLATE 8 LSB'S
01F16 9230F AF7FF 19087              WRTL HSSMA,LDMDAT,R0 & RTS         ;WRITE A BYTE & RETURN
```

Addr Line CONVERT DISK HEADER TO COUNT

```
                19089   TITLE GET DEFECTIVE CYLINDER
                19090
                19091   GTDEFCYL: JSR.P SAVRET2                     ;SAVE RETURN LOC.
01F17 E0402 1F17F 19091 +         SETPGE SAVRET2_SHR_H#B & JSRP SAVRET2
01F18 3710F F00B2 19092           RESETRAM MAINILD2                 ;CLEAR ILD DISPLAY BITS 23-12
01F19 05340 F004F 19093           ADD DB,B,MSK1,R0 & NOP D#79
01F1A 0F322 E0E7B 19094           CALL R2,ISSCMDA1                  ;MSG: "ENTER DEF. CYL. NO. "
                19095             JSR.P CHCNT                       ;GET CHARACTERS EXPECTED
01F1B E0402 1F1B7 19095 +         SETPGE CHCNT_SHR_H#B & JSRP CHCNT
01F1C 3705F F0087 19096           RDMSKQ 87
01F1D 0F322 E0E7C 19097           CALL R2,ISSCMDA                   ;GET DEF. CYL. FROM OPERATOR
01F1E 3410F F00B2 19098           ALDRAM R0,MAINILD2                ;DISPLAY IT
01F1F 373F1 F0245 19099           RDRAMB R1,MAXCYLDI                ;MAX. DRIVE CYLINDER
                19100   ;
01F20 29101 FFFFF 19101           SUBB AB,FOUTL,R0,R1               ;CYLINDER NO.> MAX?
                19102             CJMP.P AGB,BADCYL                 ;YES, GET NEW ENTRY
01F21 E0402 341F6 19102 +         SETPGE BADCYL_SHR_H#B & CJMPP AGB,BADCYL
01F22 3410F F00B8 19103           ALDRAM R0,STCYL                   ;SAVE DEFECTIVE AS STARTING CYL.
01F23 FFFFF E1185 19104           JMP RTRN2                         ;RETURN
                19442   TITLE
```

```
 Addr             Line


                  19444
01F24 FFFFF 1F72A 19445    SETUMAX:  JSR SMMAX                          ;COMPUTE MAX. ADDRESS
01F25 45050 F00FF 19446              TSTRBIT R0,H#FF                    ;ISOLATE THE 8 LSB'S AND...
01F26 321FF F035F 19447              QLDRAM URDDMAL                     ;SAVE THEM FOR READ CHECKING
01F27 3491F F035C 19448              ALDRAMU R1,URDDMAU                 ;SAVE THE MSB'S FOR CHECKING
                  19449              JSR.P NXTUADDR                     ;NEXT U-ADDR. IS 0000
01F28 E0400 1F4B0 19449 +            SETPGE NXTUADDR_SHR_H#B & JSRP NXTUADDR
01F29 FFFFF E0949 19450              JMP UBOOK02                        ;RETURN TO BOOKKEEPING
                  19451
                  19452
01F2A 373F1 F02D0 19453    SMMAX:    RDRAMB R1,MAXREL                   ;GET MAX. RELATIVE RECORD...
01F2B 373F2 F02CD 19454              RDRAMB R2,RECLNTH                  ;TIMES RECORD LENGTH...
01F2C 34354 1F568 19455              RTOR R5,R4 & JSR MULTPLYD          ;EQUALS MAX. RELATIVE WORD...
01F2D 34345 E0368 19456              RTOR R4,R5 & JMP FOUR5DN           ;TIMES 4.5 = MAX. RELATIVE BYTE
                  19457
01F2E 4505E F0200 19471    SMERR6A:  TSTRBIT R14,200                    ;DEFECTIVE TRACK FLAG?
                  19472              CJPOP.P NALUZ,SMERR33              ;YES, SET ECC CORRECTABLE
01F2F E0400 B3E9C 19472 +            SETPGE SMERR33_SHR_H#B & CJPOPP NALUZ,SMERR33
01F30 373F1 F00BA 19473              RDRAMB R1,ILDFLG                   ;GET ILD FLAGS
01F31 45051 F0400 19474              TSTRBIT R1,400                     ;FORMAT DEFECTIVE?
01F32 FFFFF A3FFF 19475              CRTS NALUZ                         ;YES, RETURN
                  19476              JPOP.P SMERR32A                    ;ELSE, CHECK FOR 1 RETRY
01F33 E0400 BF69A 19476 +            SETPGE SMERR32A_SHR_H#B & JPOPP SMERR32A
                  19477    ;
                  19478    ;         Prep. Track
                  19479
                  19480    PREP00:   EQU $
01F34 651F7 F02CD 19484              XOR DB,FOUTL,RAME,R7 & NOP RECLNTH ;IF WORD PREP. LENGTH NOT...
                  19485              BNZ.P FMTBAD0                      ; = REC. LENGTH, ERROR
01F35 E0400 33918 19485 +            SETPGE FMTBAD0_SHR_H#B & BNZP FMTBAD0
01F36 4505E F0080 19486              TSTRBIT R14,80                     ;CHECK 1-TRACK FLAG
01F37 FFFFF 33F3C 19487              BNZ PREP02                         ;IF SET, JUMP
01F38 370FF F02D7 19489              RDRAM0 CYLDIVAM
01F39 D0FF7 FFFFF 19490              RDXFR SMC,,R7
01F3A 357F7 F02D8 19491              OR DB,LSBZ,RAME,R7 & NOP CYLDIVAL  ;< 1/2 WORDS PER CYLINDER TO R7
01F3B FFFFF E1F3D 19497              JMP PREP03
                  19498    ;
01F3C 377F7 F02D9 19500    PREP02:   OR DZ,LSBZ,RAME,R7 & NOP WRDSTRK   ;< 1/2 WORDS PER TRACK TO R7
                  19504    ;
01F3D 337F7 FFFFF 19505    PREP03:   OR ZB,LSBZ,,R7                     ;FULL PREP. LENGTH TO R7
```

```
 Addr             Line

01F3E 3735F F0200 19506            RDMSKB R15,200                    ;SET FORMAT FLAG
01F3F 433F6 E053D 19507            RESETR R6 & JMP WRT000+1          ;WRITE ZEROS DATA
                  19510    ;       Swallow-5 Rev. J/K code
                  19511
01F40 4505F F2000 19512    REVJK:  TSTRBIT R15,2000                  ;Polling flag?
01F41 FFFFF 33744 19513            BOZ REVJKB                        ;No, jump
01F42 45058 F0100 19514    REVJKA: TSTRBIT R8,JDRIVE                 ;Type J drive?
01F43 320FF E1F4E 19515            OR ZQ,Q & JMP REVJKC              ;Jump & check for it
                  19516    ;
01F44 45055 F0040 19517    REVJKB: TSTRBIT R5,BUSY                   ;Drive busy?
01F45 FFFFF 33F42 19518            BNZ REVJKA                        ;Yes, can't read drive options
01F46 4505D F0070 19519            TSTRBIT R13,70                    ;ISOLATE CARD CAGE SELECT IN Q
01F47 36345 F0008 19520            OR DQ,B,MSK1,R5 & NOP H#8         ;ADD UNIT SELECT BIT 3
01F48 91E5F FFFFF 19521            WRTL HSSM,LDDRUSEL,R5
                  19522            JSR.P RDSECT1                     ;READ DRIVE OPTIONS
01F49 E0403 1F491 19522 +          SETPGE RDSECT1_SHR_H#B & JSRP RDSECT1
01F4A 323F5 FFFFF 19523            QTOR R5                           ;CARD CAGE SELECT TO R5
01F4B 91E5F FFFFF 19524            WRTL HSSM,LDDRUSEL,R5             ;CLEAR UNIT SELECT BIT 3
01F4C 55358 F0100 19525            RSTRBIT R8,JDRIVE                 ;CLEAR TYPE 'J' FLAG
01F4D 65151 F0437 19526            TSTREQU R1,VALSECT+H#37           ;DRIVE TYPE 'K'?
01F4E 37344 A32DA 19527    REVJKC: RDREGB MSK1,R4 & CRTS ALUZ,D#730  ;Yes, Useable Cylinder = 730
                  19528    ;
01F4F 0C14F F039C 19529            ADD AZC,FOUTL,R4,RAME & NOP FRSTALT ;First Alternate Cyl.= 731
                  19530            LDAR.P SEL14A+1                   ;JUMP LOC.
01F50 E0401 5F26F 19530 +          SETPGE SEL14A+1_SHR_H#B & LDARP SEL14A+1
01F51 05053 F1F55 19531            ADD DB,Q,MSK2,R3 & NOP $+D#4      ;COMPUTE JUMP LOC...
01F52 321FE FFFFF 19532            QLDREG IREG                       ;ACCORDING TO PREP. LENGTH
01F53 37929 FFFFF 19533            OR DZ,FOUTU,SEQAR,IREGM
01F54 35348 6F100 19534            OR DB,B,MSK1,R8 & JIR JDRIVE      ;SET TYPE 'J' FLAG
01F55 37344 AF000 19535            RDREGB MSK1,R4 & RTS D#0          ;INVALID PREP. LENGTH
01F56 37344 AF249 19536            RDREGB MSK1,R4 & RTS D#585        ;MAX. USEABLE CYL. @ 112 = 585
01F57 37344 AF209 19537            RDREGB MSK1,R4 & RTS D#521        ;MAX. USEABLE CYL. @ 448 = 521
01F58 37344 AF209 19538            RDREGB MSK1,R4 & RTS D#521        ;MAX. USEABLE CYL. @ 1792= 521
                  19540
                  19541
                  19542    ;    THIS ROUTINE WILL WRITE ONE RECORDS WORTH OF DATA TO THE MDB STARTING AT
                  19543    ;LOC. 0000.  THE DATA MUST BE PREVIOUSLY STORED IN FORDAT3, FORDAT2 & FORDAT1.
                  19544    ;REGISTERS USED:  R0-R3,R4,R6 & Q.
                  19545    ;
                  19546    FORMDAT: LDWD R3,FORDAT2                  ;Get bits 0-23
01F59 373F3 F026B 19546 +          RDRAMB   R3,FORDAT2               ;GET 12 MSBS
01F5A D0FF3 FFFFF 19546 +          RDXFR    SNC,,R3                  ;PUT IN UPPER REGISTER
01F5B 353F3 F026C 19546 +          OR DB,B,RAME,R3 & NOP FORDAT2+1   ;OR IN 12 LSBS
01F5C 373F6 F02CD 19547            RDRAMB R6,RECLNTH                 ;GET THE RECORD LENGTH
01F5D 433F4 FFFFF 19548            RESETR R4                         ;SET START WORD = 0
                  19549    ;
01F5E 370FF F026A 19550    FORMDAT1: RDRAMQ FORDAT3                  ;GET DATA BITS 24-35
01F5F 34340 1F68F 19551            RTOR R4,R0 & JSR MDBWRT           ;WORD NO. TO R0; WRITE ONE WORD
01F60 133F6 FFFFF 19552            DECRB R6                          ;DECR. THE WORD COUNT
```

```
 Addr             Line

01F61 0B3F4 33F5E 19553            INCRB R4 & BNZ FORMDAT1          ;IF WORD COUNT NOT 0, WRITE ANOTHER
01F62 FFFFF AF7FF 19554            RTS                              ;ELSE, RETURN
                  19555
01F63 37352 F008D 19556   RESEEK: RDMSKB R2,SKERRBIT               ;RESET THE SEEK...
                  19557            JSR.P DELABIT                    ;ERROR BIT FOR THIS DRIVE
01F64 E0401 1F1DC 19557 +          SETPGE DELABIT_SHR_H#B & JSRP DELABIT
01F65 0F10F F0302 19558            ADD DZC,FOUTL,ZERO,RAME & NOP SEEKONL;Set the Seek Only flag
01F66 433F9 FFFFF 19559            RESETR R9
                  19560            JSR.P HEADRD                     ;GET SAVED HEAD THIS DRIVE
01F67 E0400 1F1F1 19560 +          SETPGE HEADRD_SHR_H#B & JSRP HEADRD
01F68 3432C FFFFF 19561            RTOR R2,R12                      ;MOVED TO R12
                  19562            JSR.P ADVHEAD                    ;SET THE HEAD
01F69 E0400 1F51C 19562 +          SETPGE ADVHEAD_SHR_H#B & JSRP ADVHEAD
01F6A 3535F F0088 19563            SETRBIT R15,88                   ;SET SEEK ONLY & RTZ FLAGS
                  19564            JSR.P CYL00                      ;LOC. OF CUR. CYL. THIS DRIVE
01F6B E0400 1F1E9 19564 +          SETPGE CYL00_SHR_H#B & JSRP CYL00
01F6C 373DB E0149 19565            RDREGB RAM1,R11 & JMP SEEK04     ;GET THE CYLINDER & RE-SEEK
                  19566   ;
                  19573   ;================================================================
                  19574   TITLE GET DRIVE STATE
```

```
 Addr            Line GET DRIVE STATE


02000            19577    ORG H#2000
                 19578
                 19580    ;┌──────────────────────────────────────────────────────────────────────
                 19581    ;| PROCEDURE NAME: DRIVSTAT          (get_drive_state)
                 19582    ;|
                 19583    ;| PURPOSE:
                 19584    ;|    This procedure is used to get the drive state flags from the
                 19586    ;|$   OCP shared RAM.
                 19590    ;|
                 19591    ;| USAGE:
                 19592    ;|    This procedure is called on the stack (JSR) from the various
                 19593    ;|    routines which require the drive state flags.
                 19594    ;|
                 19595    ;| INPUT PARAMETERS:
                 19596    ;|    R1 = logical unit number
                 19597    ;|
                 19598    ;| CONTROL PARAMETERS:
                 19599    ;|
                 19600    ;|    drive_state_values:
                 19601    ;|         0 = illegal value
                 19602    ;|         1 = drive is DOWN
                 19603    ;|         2 = drive in CACHE mode
                 19604    ;|         3 = drive in BYPASS mode
                 19605    ;|         4 = drive in READ ONLY mode
                 19606    ;|         5 = drive in WRITE THRU mode
                 19607    ;|       6-FF = illegal values
                 19608    ;|
                 19609    ;| OUTPUT PARAMETERS:
                 19610    ;|    R8 = drive state if no error is indicated
                 19612    ;|$or R8 = 3 (BYPASS) if there is an error
                 19614    ;|    Qreg = 0, ALUZ = true, drive state cannot be returned
                 19615    ;|    Qreg > 0, ALUZ = false, drive state in R8 is valid.
                 19616    ;|    R1 = logical unit number
                 19617    ;|
                 19618    ;| REGISTERS USED:
                 19619    ;|    R0,R1,R4,R13,Q_reg & IREG.
                 19620    ;└──────────────────────────────────────────────────────────────────────
                 19621
02000 3710F F00A6 19622    DRIVSTAT: RESETRAM PARCNT                ;RESET PARITY RETRY COUNT
                 19623    ;
                 19624    ;CHANGE LOGICAL TO PHYSICAL DRIVE NO. IN R13
                 19625    ;
02001 3431D FFFFF 19626              RTOR     R1,R13                 ;Requested unit to R13
02002 45351 F001F 19627              MSKRBIT  R1,1F                  ;Isolate drive number
02003 3411F F0261 19628              ALDRAM   R1,REQUNIT             ;Save requested unit
                 19629               JSR.P    DOUBLE                ;DOUBLE IT IN CASE OF ERROR
02004 E0403 1F125 19629 +            SETPGE DOUBLE_SHR_H#B & JSRP DOUBLE
```

Addr Line GET DRIVE STATE

```
                19630   ;
                19631   ;SELECT THE OCP TO GET DRIVE STATE VALUE
                19632   ;
02005 37350 F2009 19633   DRVSTAT0: RDMSKB R0,DRVSTATA              ;Return loc. from OCP select
02006 3410E C0002 19634           ALDREG R0,IREG & LDCT 2         ;timeout is 3 micro_seconds
02007 34909 FFFFF 19636           ALDREGU  R0,IREGH               ;$return address to IREG
02008 FFFFF E0E17 19641           JMP SELOCPA                     ;Select the OCP
                19642   ;
02009 433F0 3381A 19644   DRVSTATA: RESETR   R0 & BNZ DRVSTAT7     ;$If not selected, BYPASS
                19648   ;
                19649   ;READ DRIVE_STATE FROM OCP
                19650   ;
0200A 373F1 F0261 19651   DRVSTAT1: RDRAMB R1,REQUNIT              ;GET REQUESTED UNIT NO.
0200B 9701F 4F001 19652           WRTL OCP,OCPADR,R1 & PSHLD 1    ;OUTPUT THE UNIT NUMBER
0200C C7F18 887FF 19653           RD OCP,OCPDATA,R8 & CLPCNT EVCNTZ
0200D C7F18 4F002 19654           RD OCP,OCPDATA,R8 & PSHLD 2     ;READ DRIVE STATE AND...
0200E 335F8 887FF 19655           RSHFTZ R8                       ;MOVE PRESENT STATE INTO POSITION
0200F 45358 F000F 19656           MSKRBIT R8,H#F                  ;MASK OUT ANY JUNK BITS
                19657   ;
                19658   ;CHECK IF DATA IS RELIABLE
                19659   ;
02010 37364 FFFFF 19660           RDREGB SMCST,R4                 ;READ & RESET XMC STATUS
02011 45154 F0004 19661           TSTBIT R4,4                     ;IF CT-BUS PARITY ERROR?
02012 FFFFF 33818 19662           BNZ DRVSTAT3                    ;YES, possible retry
                19663   ;
                19664   ;CHECK FOR OCP MEMORY PARITY ERROR
                19665   ;
02013 C7F01 4F001 19666           RD OCP,OCPSTAT,R1 & PSHLD 1
02014 C7F01 887FF 19667           RD OCP,OCPSTAT,R1 & CLPCNT EVCNTZ ;READ OCP STATUS
02015 45151 F0100 19668           TSTBIT R1,100                   ;OCP MEMORY PARITY ERROR?
02016 E7800 33818 19669           CTL OCP,OCPRST & BNZ DRVSTAT3   ;YES! CHECK IF RETRY EXHAUSTED
                19670           ADD     DZC,0,SMCST             ;SET DATA VALID STATUS
02017 0F06F E201B 19671           /&      JMP DRVSTAT9            ;...AND return
                19672   ;
                19673   ;CT-BUS OR OCP_MEMORY PARITY ERROR
                19674   ;
02018 34300 FFFFF 19675   DRVSTAT3: RTOR     R0,R0                ;TEST RETRY COUNT
                19676           INCRB   R0                      ;INCREMENT RETRY COUNT
02019 0B3F0 3300A 19677           /&      BUZ DRVSTAT1            ;...AND RETRY ONCE IF = 0
                19678
                19700   ;
                19701   ;RETURN WITH DRIVE_STATE = BYPASS
                19702   ;
0201A 37358 F0003 19703   DRVSTAT7: RDMSKB   R8,H#3                ;SET DRIVE STATE = BYPASS
```

Addr Line GET DRIVE STATE

```
                19704   ;
                19705   ;EXIT WITH ENDING STATUS
                19706   ;
0201B E7C00 FFFFF 19707   DRVSTAT9: CTL     OCP,UCPDSEL             ;RELEASE THE OCP
0201C 320FF AF7FF 19708            QTOQ    & RTS
                19709
                19710   ;================================================================
                19711   TITLE
```

```
 Addr             Line


                  19801
0201D C10C0 FFFFF 19802    SNTlMERA: RD HSSM,RDDREC,R0
0201E C10C0 FFFFF 19803              RD HSSM,RDDREC,R0                ;READ THE REC. NO. (ERROR CODE)
0201F 65050 F00E0 19804              TSTRMSK R0,H#E0                  ;DID MPC TAKE A BAD JUMP?
02020 FFFFF 33022 19805              BOZ $+2                          ;YES, CATASTROPHIC ERROR, JUMP
02021 E1000 E0682 19806              CTL HSSM,DIFRST & JMP SMERR25    ;ELSE, NORMAL RETRY
                  19807    ;
                  19808              JSR.P CLRHSSM                    ;CLEAR THE HSSM BOARDS AND...
02022 E0400 1F340 19808 +            SETPGE CLRHSSM_SHR_H#B & JSRP CLRHSSM
02023 FFFFF E2023 19809              JMP $                            ;HANG THE MICROCODE
02024 FFFFF E02C0 19826    MULTPLYE: JMP MULTPLY
                  19827
                  19829    ;$    CDC DOES NOT SUPPORT LOG OVERFLOW.  WHENEVER ANY ERROR/USAGE LOG
                  19830    ;$OVERFLOWS, THIS ROUTINE WILL CLEAR THE ERROR/USAGE LOG FOR THIS DRIVE AND
                  19831    ;$RETURN.
                  19836
                  19837    ;OVFLWBIT: EQU H#90                        ;ERROR/USAGE OVERFLOW
                  19838    ;
                  19840    LOGOFL:   JSR.P ERRLOG                    ;$LOC. OF ERROR/USAGE LOG
02025 E0400 1F350 19840 +            SETPGE ERRLOG_SHR_H#B & JSRP ERRLOG
02026 3412E C000B 19841              ALDREG R2,1REG & LDCT 8          ;$MOVE IT TO THE 1REG
02027 FFFFF E033D 19842              JMP CLR25                        ;$CLEAR THE LOG THIS DRIVE & RETURN
```

```
 Addr             Line


                  19849
                  19850    ;    This routine is used in the Wait Loop to determine if Backscan Seek
                  19851    ;and ILD's should be skipped because interrupts are pending to the host(s).
                  19852    ;    If ALU = 0 on return, the Wait Loop will continue.
                  19853    ;    If ALU not = 0, Backscan Seek and the ILD parts of the Wait Loop will
                  19854    ;be skipped.
                  19855    ;
02028 3715E F02B5 19856    WAITSKIP: MOVE MSK2,1REG & NOP DRONCYL       ;Loc. of Drive on-cyl. bits
                  19857              JSR.P SMBRD                        ;Get them
02029 E0400 1F03B 19857 +            SETPGE SMBRD_SHR_H#B & JSRP SMBRD
0202A 30100 AF7FF 19861              OR AQ,FOUTL,R0,ZERO & RTS          ;Check the bits & return
```

```
 Addr             Line

                  19933    ;
                  19934    ;    THIS ROUTINE WILL SET AN ATTENTION PENDING FOR THE CURRENT DRIVE
                  19935    ;ON ALL CHANNELS.
                  19936    ;    UPON ENTRY:  R3 = BIT FOR THIS DRIVE (24-31)
                  19937    ;                 R4 = BIT FOR THIS DRIVE (0-23)
                  19941    ;REGISTERS USED:  R0,R1,R2,Q & IREG
                  19943    ;
                  19944    SETATTN:   EQU   $
02028 37352 F0072 19953             RDMSKB R2,ATTN1                 ;LOC OF ATTN. PENDING HIO-1
0202C 37160 4F003 19954             MOVE SMCST,ZERO & PSHLD 3       ;ATTENTION IS ON ALL HIO'S
                  19955             JSR.P ADDABIT                   ;ADD THE BIT FOR THIS DRIVE
0202D E0400 1F1FF 19955 +           SETPGE ADDABIT_SHR_H#B & JSRP ADDABIT
0202E 05342 88006 19956             ADD DB,B,MSK1,R2 & CLPCNT EVCNTZ,H#6 ;LOC. FOR NEXT HIO
0202F FFFFF AF7FF 19957             RTS                             ;RETURN
                  19958
                  19959
                  20509    ;
                  20510    ;        THIS ROUTINE SELECTS THE OCP.
                  20511    ;IF THE OCP CANNOT BE SELECTED ON THE FIRST TRY, IT WILL BE ATTEMPTED 5 MORE
                  20512    ;TIMES.  IF THIS FAILS THE ROUTINE WILL RETURN (RTS) WITH ALU NOT = 0.  IF
                  20513    ;SELECTION IS SUCCESSFUL, ALU = 0 UPON RETURN.
                  20514    ;
                  20515    ;REGISTERS USED: R0,R2 & IREG.
                  20516    ;
02030 FFFFF E0038 20526    SMBRDC:   JMP SMBRD
02031 FFFFF E0044 20527    SMBWRTC:  JMP SMBWRT
                  20528
02032 37352 F0005 20529    OCPSELA:  RDMSKB R2,5                     ;RETRY COUNT = 5
02033 0F320 E0E15 20530             CALL R0,SELOCP                  ;SELECT THE OCP
02034 333F2 33836 20531             OR ZB,B,,R2 & BNZ OCPSEL1       ;IF NOT SELECTED, JUMP
02035 3710F AF267 20532             MOVE ZERO,RAME & RTS OCPDOWN    ;RESET OCP DOWN & RETURN; ALU = 0
                  20533    ;
02036 133F2 33834 20534    OCPSEL1:  DECRB R2 & BNZ OCPSELA+2        ;DECR. RETRY COUNT & TRY AGAIN
02037 0F10F AF267 20535             ADD DZC,FOUTL,ZERO,RAME & RTS OCPDOWN ;RETURN WITH ALU NON-ZERO
                  21089
02038 C1043 FFFFF 21090    CKONCYL:  RD HSSM,RDDRST1,R3
02039 C1043 FFFFF 21091             RD HSSM,RDDRST1,R3              ;READ CURRENT DRIVE STATUS
0203A 55353 F0010 21092             RSTRBIT R3,WRITPROT             ;DON'T CHECK WRITE PROTECT
0203B 65053 F00A3 21093             TSTRMSK R3,SKEND+SELECT+ONCYL+READY ;NORMAL STATUS?
                  21094             BOZ.P CHK2NDA                   ;YES, CHECK FOR SECOND DRIVE
```

```
 Addr             Line

0203C E0403 33192 21094 +          SETPGE CHK2NDA_SHR_H#B & BOZP CHK2NDA
                  21095            JSR.P SEEK30                    ;WAIT FOR NORMAL STATUS AND...
0203D E0400 1F18B 21095 +          SETPGE SEEK30_SHR_H#B & JSRP SEEK30
0203E FFFFF E1992 21096            JMP CHK2NDA                     ;CHECK FOR SECOND DRIVE
                  21097
                  21294
                  21295    TITLE FUNCTION & STATUS TRACING
```

```
 Addr            Line FUNCTION & STATUS TRACING


                 21297
                 21298  ;         (OPTIONAL FUNCTION AND STATUS TRACE CODE)
                 21299  ;
0205D            21301  ORG H#205D        ;$for CDC
                 21305  ;
0205D 373F2 F002D 21306  EFTRACE:  RDRAMB R2,EFALLOW                ;GET THE BIT FOR THIS HAIO
0205E 01722 4F002 21307           ADD AB,LSBZ,R2,R2 & PSHLD 2      ;LEFT SHIFT IT...
                 21308            LSHFT2 R2                        ;EIGHT PLACES
0205F 01722 887FF 21308 +         ADD AB,LSBZ,R2,R2 & CLPCNT EVCNTZ
                 21309  ;
02060 313C2 FFFFF 21310           OR AB,B,R12,R2                   ;DISPLAY HAIO & COMMAND
02061 333FD E007A 21311           OR ZB,B,,R13 & JMP EFPROCC       ;DISPLAY DRIVE NUMBER
                 21312  ;-------------------------------------------------------------------------
02062 31321 FFFFF 21313  EITRACEA: OR AB,B,R2,R1                   ;DISPLAY HAIO & STATUS ACTION
02063 333FD E0DB0 21314           OR ZB,B,,R13 & JMP END53A        ;DISPLAY STATUS & DRIVE NO.
                 21315  ;
02064 373F1 F002D 21316  EITRACE:  RDRAMB R1,EFALLOW                ;GET THE BIT FOR THIS HAIO
02065 01711 4F002 21317           ADD AB,LSBZ,R1,R1 & PSHLD 2      ;LEFT SHIFT IT...
                 21318            LSHFT2 R1                        ;EIGHT PLACES
02066 01711 887FF 21318 +         ADD AB,LSBZ,R1,R1 & CLPCNT EVCNTZ
02067 345A2 FFFFF 21319           OR AZ,RSBZ,R10,R2                ;RIGHT SHIFT STATUS ACTION...
02068 335F2 E2062 21320           OR ZB,RSBZ,,R2 & JMP EITRACEA    ;TWO PLACES & JUMP
                 21321


                 21501  TITLE CDC O.S. CROSS REFERENCE
                    21509       END ;          ;Cross Refernce is Condensed Elite printing
```

Assembly Phase complete.
0 error(s) detected.