Inter-Office Memorandum

To	F Ludolph	Date	February 13, 1978
From	JR Cucinitti	Location	Palo Alto
Subject	Disk Diagnostic	Organization	SDD/CS/SD

XEROX

Filed on: <Cucinitti>Diskdiag.memo

The Disk Diagnostic pack has two diagnostics written in four different places in anticipation of loosing one or more of them. One diagnostic is booted by using just the boot button in the first case, and depressing the 0 key and pushing the boot button in the next. The other diagnostic is booted with the K key and boot button, and the V key and the boot button.

The diagnostic booted with no keys and the 0 key will write one pass on the disk and then start into a read loop, we call it write once, read forever. To run the diagnostic just type 40001B (this sets an error breakpoint), 40000P (sends the Alto to 40000 to fetch the first instruction). The diagnostic booted with the K or V key will write, read, write, read,..... To run this diagnostic type 1001B (sets the error breakpoint), 1000P (fetches the instruction from 1000). To halt the diagnostic just depresss the Swat key, this will display four registers, to continue depress the P (procede) key.

The knowledge of the location of the diagnostics in the Alto memory is of some use. The diagnostic booted by the 0 key resides from 40000 thru 42154 and uses memory locations from 1000 thru 37777 for the data handling area. The diagnostic booted by the K key resides from 1000 thru 3153 and uses 40000 thru 176777 for the data handling area. The first 777 locations are used to run the display and keyboard and should not be changed. You may note that the listing does not reflect the true memory locations but do indicate the locations relative to the starting locations of the diagnostics.

The debugger located in the first 1000 locations will allow you to modify any memory location and remember it only knows about octal numbers. The commands to the debugger are:

n/ opens and displays memory location *n*
cr inserts the typed information in *n* and closes the location
← modifies, closes location *n*, and opens location *n*+1
↑ modifies, closes location *n*, and opens location *n*-1
A displays accumulators 0-3
*n*B set a breakpoint at location *n*
*n*D deletes breakpoint n (1-9)
*n*P procede from this location

When you do get an error you will see something like this:

DATA COMPARE FAILED
0:123456 1:123455 2:012345 3:001612

AC0 = Data that was to be written on the disk
AC1 = Data that was read from the disk (this may indicate a memory problem)
AC2 = Memory location the disk data was written into
AC3 = Of no intrest concerns the text on the display

CONTROLLER REPORTS BAD STATUS:
0:002310 1:007541 2:034523 3:001612

AC0 = Of no intrest
AC1 = Disk controller status
AC2 = Pointer to disk control block that failed
AC3 = Of no intrest concerns the text on the display

See the Alto hardware manual for the orginization of the disk command blocks.

To continue from an error breakpoint just type P.

Some of the intresting locations in the diagnostic:
0562 Current sector (0-13) updated by the diagnostic
0563 Minimum sector number
0564 Maximum sector number
0565 Current head number (0=upper, 1=lower)
0566 Minimum head number
0567 Maximum head number
0570 Current drive (0 or 1)
0571 Minimum drive number
0572 Maximum drive number
0573 Current track number
0574 Minimum track (never less than 100)
0575 Maximun track

These locations may be changed to write only with one head and just in one sector on one track. If 0574 is changed you may write over the boot loader so **never** have it less than **100.**

Some samples of the disk command blocks that are used for alignment:

Four cylinder seek

1000/2000	2000/1000
1001/0	2001/0
1002/44002	2002/44002
1003/0	2003/0
1004/0	2004/0
1005/0	2005/0
1006/0	2006/0
1007/0	2007/0
1010/0	2010/0
1011/10	2011/50

At this time set location 521 to 1000, when this location has some value other than 0 the disk microcode will go to that location for the disk command block. To halt the loop set location 521 to 0.

Track 0 or restore adjustment

1000/2000	2000/1000
1001/0	2001/0
1002/44002	2002/44002
1003/0	2003/0
1004/0	2004/0
1005/0	2005/0
1006/0	2006/0
1007/0	2007/0
1010/0	2010/0
1011/0	2011/51

For head alignment set the command block as follows:

1000/1000
1001/0
1002/44002
1003/0
1004/0
1005/0
1006/0
1007/0
1010/0
1011/1510(upper head), 1011/1514(lower head)

The data burst is adjusted with the same block by changing only location 1011.
1011/1440(upper head), 1011/1444(lower head)

The disk address may be changed at any time and the microcode will ddo the right thing, so you need not stop the disk (521/0) to switch from one head to the other.

This is the listing of the diagnostic, if the diagnostic is loaded at 1000 then the first address is not 0 but 1000 and the listing reflects the locations relative to 1000, if loaded at 40000 all locations are relative to 40000.

```
0001  ALTKD
```

```
                ;
                ;
                ;   ALTO DISK DIAGNOSTIC
                ;
                ;       E. MCCREIGHT
                ;
                ;
102513      .DALC   SG=SUBL# 0,0,SNC
102512      .DALC   SIZE=SUBL# 0,0,SZC
                .TITL   ALTKD
                .NREL
                .EXTN PHIAD,MESSAGE
000001      .TXTM 1
                ;
                ;   ALTO CONSTANTS
                ;
000521 KBCON=521;   DISK BLOCK ADDRESS
000420 DASCON=420;  DISPLAY BLOCK ADDRESS
000452 WWCON=452;   WAKEUPS WAITING ADDRESS
000501 IVCON=501;   VECTOR OF INTERRUPT NEW PC'S
061001 EIR=61001;   OPCODE FOR ENABLE INTERRUPTS
061000 DIR=61000;   OPCODE FOR DISABLE INTERRUPTS
061002 BRI=61002;   OPCODE FOR RETURN FROM INTERRUPT
000527 TRPC=527;    BEFORE A TRAP THE PC GOES HERE
000530 TRPVEC=530;  THE VECTOR OF 37 TRAP LOCATIONS GOES HERE
062000 ERROR=62000; UNIMPLEMENTED I/O OPERATION WHICH TRAPS
                ;
                ;   THE OLD FORMAT OF A DISK CONTROL BLOCK
                ;
000000 POINTER=     0
000001 STATUS=      1
000002 DSKCOMM=     2
000003 LABEL=       3
000004 DATA=        4
000005 HDRWD=       5
000006 DSKADR=      6
000007 SUCCESS= 7
000010 FAILURE=     10
000011 HEADER=      11
000012 FINISHUP=    12
000013 INTSDONE=    13
000014 SERRS=       14
000015 DCBEND=      15


                ;
                ;   INITIALIZE THE DISK
                ;
00000'000402 INIT:  JMP     .+2;SKIP OVER ERROR BREAK

00001'002411 BREAK:     JMP@    .RR2;   PUT DEBUGGER BREAKPOINT HERE !!!

00002'006401    JSR@    .+1;SET UP THE INTERRUPT SYSTEM
00003'001500'   SIZEM
00004'126400    SUB     1,1;  ZERO THE ERROR COUNTER
00005'000406    JMP     INITA;  SKIP OVER THE DISPLAY CONTROL BLOCK
00006'000000 DB:    0;      MUST BE AN EVEN ADDDRESS
0002  ALTKD
```

```
00007'000040    40
00010'000000 DBMP:      0
00011'000536    350.

00012'001473'.RR2:   RR2

00013'044011-INITA:     STA     1,SERRCNT;  ZERO SOFT ERROR COUNT
00014'044012-   STA     1,HERRCNT;          AND HARD ERROR COUNT
00015'022433    LDA@    0,.RBOT;        REASON TABLE RESET TO EMPTY
00016'042433    STA@    0,.RTOP
00017'006401    JSR@ .+1
00020'001042'   SUINT
00021'030004-   LDA     2,KBLKADR;      SET UP ILLEGAL CURRENT ADRESS TO
00022'020425    LDA     0,ALLONES;  FORCE A SEEK
00023'041002    STA     0,2,2
00024'030003-   LDA     2,DASTART
00025'020425    LDA     0,DBLKAD
00026'041000    STA     0,0,2;          START THE DISPLAY
00027'006424    JSR@    SDBAD;      SET UP DISK CONTROL BLOCK CHAINS
00030'101010    MOV#    0,0;        NOP
00031'002401 REDO:      JMP@    .+1
00032'000033'   WRTIT

00033'006456 WRTIT:     JSR@    SMAD
00034'000446    JMP     IDONE
00035'030417 IMORE:     LDA     2,WRTBLK;   SET UP TO WRITE AT CURRENT DISK ADDRESS
00036'004406    JSR MKDCB
00037'101010    MOV#    0,0;        NOP
00040'006453    JSR @ARGEN;         INITIALIZE DATA BLOCK
00041'101010    MOV#    0,0;        NOP
00042'006401    JSR@    .+1; ENTER CPTR IN COMMAND QUEUE
00043'000414'   EQUE
00044'006446    JSR@    IAD;            GET THE NEXT DISK ADDRESS
00045'000415    JMP     RDIT;           IF NONE EXIST, READ THE RECORDS
00046'000767    JMP     IMORE;          OTHERWISE, DO IT AGAIN
00047'177777 ALLONES:       177777
00050'000356'.RBOT:     RSNBOT
00051'000400'.RTOP:     RSNTOP
00052'000006'DBLKAD:    DB
00053'000775'SDBAD:     SUDB
00054'000055'WRTBLK:    .+1
00055'000106'   WRTALL
00056'000107'   IHDR
00057'000001-   CODAD
00060'000110'   NOPAD
00061'000601'ANOERR:    NOERR


                ;
                ;   READ ALL RECORDS ON THE DISK AND COMPARE THEM WITH WHAT
                ;   WAS WRITTEN
                ;

00062'006427 RDIT:      JSR@    SMAD;   GO BACK TO THE MINIMUM DISK ADDRESS
00063'000417    JMP     IDONE;  (IF ERROR)
00064'030406 RDMORE:        LDA     2,RDBLK;    READ A BLOCK AT THE CURRET ADDRESS
00065'004427    JSR DODCB
00066'101010    MOV#    0,0;  NOP
00067'006423    JSR@    IAD;    GET NEXT DISK ADDRESS
0003  ALTKD
```

```
00070'000412    JMP     IDONE;  IF WE ARE AT THE END, EMPTY THE QUEUE
00071'000773    JMP     RDMORE;    OTHERWISE GET A NEW QUEUE ENTRY AND TAKE OFF

00072'000073'RDBLK:     .+1
00073'000077'   CHRR
00074'000101'   SUBST
00075'000001-   CODAD
00076'000100'   ACDAT


00077'000120 CHRR:      120; CHECK HEADER AND LABEL, READ REST OF SECTOR
00100'000615'ACDAT:     CDAT;
00101'000000 SUBST:     0


00102'006401 IDONE:     JSR@    .+1;MAKE SURE COMMAND QUEUE EMPTIES
00103'000464'   FLUSH
00104'002401 SMASH:     JMP@    .+1
00105'000031'   REDO


00106'000374 WRTALL:    374
00107'171717 IHDR:      171717
00110'000602'NOPAD:     CKDERR;
00111'000474'SMAD:      SETMIN
00112'000506'IAD:    INCAD
00113'000650'ARGEN:     RGEN


                ;
                ;   MAKE UP A DISK CONTROL BLOCK AND QUEUE IT FOR THE DISK
                ;
00114'054407 DODCB:     STA     3,R3DOD
00115'004407    JSR MKDCB;      MAKE UP THE PROPER DCB
00116'101010    MOV#    0,0;  NOP
00117'006401    JSR@    .+1;QUEUE IT
00120'000414'   EQUE
00121'034402    LDA     3,R3DOD
00122'001401    JMP     1,3

00123'000000 R3DOD:     0


                ;
                ;   MAKE UP A DISK CONTROL BLOCK. REGISTER 2 CONTAINS A POINTER
                ;   TO A FOUR-WORD BLOCK OF POINTERS TO ITEMS TO BE INCLUDED
                ;   IN THE CONTROL BLOCK: (0) THE DISK COMMAND
                ;       (1) THE HEADER WORD
                ;       (2) THE DISK ADDRESS
                ;       (3) THE FINISHUP ROUTINE
                ;
00124'054422 MKDCB:     STA     3,R3MKD
00125'050422    STA     2,MKDPARM
00126'004422    JSR GCB;    GET AN AVAILABLE COMMAND BLOCK
00127'101010    MOV#    0,0;  NOP
00130'030002-   LDA     2,CPTR;  THE COMMAND BLOCK
00131'034416    LDA     3,MKDPARM
00132'023400    LDA@    0,0,3;     THE DISK COMMAND
00133'041002    STA     0,DSKCOMM,2
0004  ALTKD
```

```
00134'023401    LDA@   0,1,3;     THE HEADER WORD
00135'041005    STA    0,HDRWD,2
00136'023402    LDA@   0,2,3;     THE DISK ADDRESS
00137'041006    STA    0,DSKADR,2
00140'023403    LDA@   0,3,3;     THE FINISHUP ROUTINE
00141'041012    STA    0,FINISHUP,2
00142'102400    SUB    0,0; MARK BLOCK AS UNPROCESSED BY INTERRUPT
00143'041013    STA    0,INTSDONE,2;    ROUTINE
00144'034402    LDA    3,R3MKD
00145'001401    JMP    1,3

00146'000000 R3MKD:    0
00147'000000 MKDPARM: 0


                ;
                ;    GET A POINTER TO AN AVAILABLE COMMAND BLOCK IN CPTR. ALWAYS
SKIPS.
                ;
00150'054414 GCB:  STA     3,R3G
00151'030005-GC1:  LDA     2,AS;     TRY TO ALLOCATE ONE FROM AVAILABLE STACK
00152'151015    MOV#   2,2,SNR;
00153'000406    JMP    STKEMP
00154'025000    LDA    1,POINTER,2;      POP THE AVAILABLE STACK
00155'044005-   STA    1,AS
00156'050002-   STA    2,CPTR
00157'034405    LDA    3,R3G
00160'001401    JMP    1,3

00161'004404 STKEMP:   JSR  GETQE
00162'000767    JMP    GC1
00163'000766    JMP    GC1

00164'000000 R3G:  0


                ;
                ;    REMOVE A FINISHED ENTRY FROM THE COMMAND QUEUE, AND PUT
                ;    IT IN THE AVAILABLE STACK. ALWAYS SKIP.
                ;

00165'054557 GETQE:    STA     3,R3Q
00166'101010 GQ1:  MOV#   0,0; NOP
00167'030006-   LDA    2,KQF;   CHECK IF COMMAND QUEUE IS EMPTY. IF SO GIVE ERROR
00170'151015    MOV#   2,2,SNR;
00171'002553    JMP@   R3Q;     DON'T SKIP IF ERROR
00172'025001    LDA    1,STATUS,2;  CHECK THE STATUS OF THE QUEUE FRONT
00173'125014    MOV#   1,1,SZR
00174'000535    JMP    PULLIT; CONTROLLER GOT TO IT
00175'034004-   LDA    3,KBLKADR;       MAKE SURE CONTROLLER IS RUNNING
00176'021400    LDA    0,0,3
00177'101004    MOV    0,0,SZR
00200'000766    JMP    GQ1;     IT'S RUNNING. RETRY.
00201'021001    LDA    0,STATUS,2;  GET THE STATUS WORD FOR KQF
00202'101014    MOV#   0,0,SZR
00203'000763    JMP    GQ1;     COMMAND AT KQF IS DONE. PROCESS IT.
00204'050007-RES:  STA     2,KQM;  (FOR THE INTERRUPT ROUTINE)
00205'051400    STA    2,0,3;     DISK CONROLLER IS STOPPED AND HAS NOT PROCESSED
00206'000760    JMP    GQ1;     THE COMMAND AT KQF. RESTART IT.

0005  ALTKD
```

```
00207'010011-AAUGH:   ISZ SERRCNT;   THERE WAS AN ERROR. ADD ONE TO SOFT
00210'101010    MOV#  0,0;      ERROR COUNT.

00211'022004-   LDA@  0,KBLKADR;      WAIT FOR CONTROLLER TO STOP
00212'101014    MOV#  0,0,SZR
00213'000776    JMP   .-2

00214'034542    LDA   3,RSNBOT;   CLASSIFY THE ERROR
00215'020563 SRCHLP:  LDA   0,RSNTOP
00216'116512    SLE 0,3
00217'000421    JMP   LOOK

00220'020557    LDA   0,RSNEND;   THIS ERROR NOT IN TABLE
00221'116513    SG  0,3
00222'000434    JMP   HARDYET;   AND NO TABLE SPACE IS LEFT, SIGH!

00223'021001    LDA   0,STATUS,2;  MAKE TABLE ENTRY FOR THIS ERROR
00224'041400    STA   0,0,3
00225'102400    SUB   0,0
00226'041401    STA   0,1,3

00227'161400    INC 3,0; MOVE UP THE TOP OF THE TABLE
00230'101400    INC 0,0
00231'040547    STA   0,RSNTOP

00232'054513    STA   3,R3SL;       A NEW ERROR STATUS IS SEEN
00233'025001    LDA   1,STATUS,2;  SHOW STATUS TO DIAGNOSTICIAN
00234'006000-   JSR@  PMSG
00235'002135'   NEWER
00236'034507    LDA   3,R3SL
00237'062000    ERROR

00240'021001 LOOK:    LDA   0,STATUS,2;  SEE IF OUR STATUS MATCHES TABLE,
00241'024507    LDA   1,NOKCON;       EXCEPT FOR SECTOR NUMBER
00242'123400    AND   1,0
00243'025400    LDA   1,0,3
00244'106400    SUB   0,1
00245'020503    LDA   0,NOKCON
00246'107405    AND   0,1,SNR
00247'000404    JMP   EFND;       YES, IT DOES
00250'175400    INC 3,3;      NO, IT DOESN'T. GO TO NEXT ENTRY.
00251'175400    INC 3,3
00252'000743    JMP   SRCHLP

00253'021401 EFND:    LDA   0,1,3;          INCREMENT NUMBER OF OCCURRENCES
00254'101400    INC 0,0
00255'041401    STA   0,1,3

00256'021014 HARDYET: LDA   0,SERRS,2;   NUMBER OF SOFT ERRORS FOR THIS BLOCK
00257'101400    INC 0,0
00260'041014    STA   0,SERRS,2

00261'024470    LDA   1,HARDTH;  HAS SOFT ERROR HARDENED?
00262'106513    SG  0,1
00263'000407    JMP   SOFT

00264'010012-   ISZ HERRCNT;   YES, SEND MESSAGE TO THAT EFFECT
00265'101010    MOV#  0,0
00266'006000-   JSR@  PMSG
0006 ALTKD
```

```
00267'002116'   HARDERR
00270'062000    ERROR

00271'000444    JMP     REMOVE;         GIVE UP ON THIS DCB AND TRY THE NEXT ONE

00272'024460 SOFT:      LDA     1,RESTH;        SHOULD WE DO A RESTORE?
00273'106513    SG  0,1
00274'000423    JMP     AGAIN;          NO, JUST RETRY.

00275'021006    LDA     0,DSKADR,2;     YES, FIGURE OUT WHICH DISK IS INVOLVED
00276'025002    LDA     1,DSKCOMM,2
00277'127000    ADD     1,1
00300'123000    ADD     1,0
00301'024452    LDA     1,DSKBIT
00302'123400    AND     1,0

00303'024451    LDA     1,RSTBIT;       COMPUTE A RESTORE DISK ADDRESS BY
00304'123000    ADD     1,0;        ADDING RESTORE BIT TO BIT FOR AFFECTED
00305'040506    STA     0,RKADR;        DISK.

00306'034004-   LDA     3,KBLKADR;      FORCE A SEEK ON NEXT OPERATION
00307'024446    LDA     1,MONE
00310'045402    STA     1,2,3

00311'020470    LDA     0,.RESTOR;  START THE DISK CONTROLLER ON A RESTORE
00312'041400    STA     0,0,3

00313'021400    LDA     0,0,3;          HAS IT STOPPED YET?
00314'101014    MOV#    0,0,SZR
00315'000776    JMP     .-2;        NO -- ASK AGAIN.

00316'045402    STA     1,2,3;          FORCE A SEEK ON NEXT OPERATION

00317'050427 AGAIN:     STA     2,R2ZAP;        SCAN THE CHAIN, RESETTING DCBS
00320'151005 ZAPLP:     MOV     2,2,SNR
00321'000405    JMP     RESTART
00322'006401    JSR@    .+1;CLEAR DCB FOR RE-ENTRY ON QUEUE
00323'001565'   SETLAB
00324'031000    LDA     2,POINTER,2
00325'000773    JMP     ZAPLP

00326'030420 RESTART: LDA       2,R2ZAP
00327'034004-   LDA     3,KBLKADR
00330'000654    JMP     RES

00331'006401 PULLIT:    JSR@    .+1; VERIFY THAT THE CORRECT INTERRUPT
00332'001442'   CKINT;      HAPPENED ON THIS BLOCK
00333'007012    JSR@    FINISHUP,2; DO THE FINISHUP ROUTINE
00334'000653    JMP     AAUGH!;     THINGS DID NOT GO SO WELL
00335'035000 REMOVE:        LDA     3,POINTER,2;        GET NEXT QUEUE FRONT ADDRESS
00336'054006-   STA     3,KQF
00337'024005-   LDA     1,AS;   PUSH THE ENTRY ON AVAILABLE STACK
00340'045000    STA     1,POINTER,2;
00341'050005-   STA     2,AS
00342'034402    LDA     3,R3Q
00343'001401    JMP     1,3

00344'000000 R3Q:   0
00345'000000 R3SL:      0
0007  ALTKD
```

```
00346'000000 R2ZAP:    0
00347'170000 OKCON:    170000
00350'007777 NOKCON:       7777
00351'000010 HARDTH:       10;  NUMBER OF SOFT ERRORS TO HARDEN
00352'000003 RESTH:    3;   NUMBER OF SOFT ERRORS TO TRIGGER RESTORE
00353'000002 DSKBIT:   2
00354'000001 RSTBIT:   1
00355'177777 MONE:     177777


00356'000357'RSNBOT:   .+1
     000020    .BLK    20;  ERRORS ARE CATEGORIZED 'N THIS TABLE:
               ;                  (0) ERROR STATUS (EXCEPT FOR SECTOR NO.)
               ;                  (1) NUMBER OF OCCURRENCES
00377'000377'RSNEND:   .
00400'000000 RSNTOP:   0


00401'000402'.RESTOR:  .+1
00402'000000   0
00403'000000   0
00404'044002   44002
00405'000000   0
00406'000000   0
00407'000000   0
00410'000000   0
00411'000000   0
00412'000000   0
00413'000000 RKADR:    0



               ;
               ;   PLACE AN ENTRY IN THE COMMAND QUEUE AND START THE
               ;   CONTROLLER IF APPROPRIATE.
               ;

00414'054456 EQUE:     STA     3,R3F
00415'030002-  LDA     2,CPTR;  THE ENTRY TO BE ADDED
00416'126400   SUB     1,1
00417'045000   STA     1,POINTER,2;     GIVE IT A 0 POINTER AND
00420'045014   STA     1,SERRS,2;   NO SOFT ERRORS
00421'020442   LDA     0,VALID;     MARK THIS AS A VALID COMMAND
00422'025002   LDA     1,DSKCOMM,2
00423'107000   ADD     0,1
00424'045002   STA     1,DSKCOMM,2
00425'006401   JSR@    .+1;CLEAR STATUS AND INTS DONE AND LABEL BLOCK
00426'001565'  SETLAB

00427'024006-  LDA     1,KQF
00430'125014   MOV#    1,1,SZR;  CHECK QUEUE FOR EMPTINESS
00431'000405   JMP     NOTEMP
00432'050006-  STA     2,KQF;   STORE THIS ENTRY IN QUEUE FRONT IF QUEUE IS EMPTY
00433'034004-  LDA     3,KBLKADR
00434'045400   STA     1,0,3;     IDLE THE DISK CONTROLLER, JUST IN CASE.
00435'000403   JMP     EQCOM
00436'034010-NOTEMP:   LDA     3,KQR
00437'051400   STA     2,POINTER,3;     POINT CURRENT REAR ENTRY TO NEW ENTRY
00440'050010-EQCOM:    STA     2,KQR;   SET REAR POINTER TO NEW ENTRY
0008  ALTKD
```

```
00441'030006-FIRELP:    LDA     2,KQF;  IS THE DISK CONTROLLER QUEUE EMPTY?
00442'151015    MOV#    2,2,SNR
00443'000416    JMP     EQDON;      IF SO, WE ARE DONE
00444'034004-   LDA     3,KBLKADR;      IS THE DISK CONTROLLER RUNNING?
00445'021400    LDA     0,0,3
00446'101014    MOV#    0,0,SZR
00447'000412    JMP     EQDON;      IF SO, WE ARE DONE
00450'021001    LDA     0,STATUS,2; CHECK THE STATUS OF THE QUEUE FRONT
00451'101014    MOV#    0,0,SZR
00452'000404    JMP     EXTRACT;    IF DONE, PULL IT OFF THE QUEUE
00453'050007-   STA     2,KQM;  (FOR THE INTERRUPT ROUTINE)
00454'051400    STA     2,0,3;      START THE DISK CONTROLLER
00455'000404    JMP     EQDON

00456'006415 EXTRACT:JSR@       .GETQE
00457'101010    MOV#    0,0;  NOP
00460'000761    JMP     FIRELP

00461'034411 EQDON:     LDA     3,R3F;
00462'001401    JMP     1,3

00463'044000 VALID:     44000

                ;
                ;   FLUSH THE COMMAND QUEUE INTO AVAILABLE SPACE AS THE CONTROLLER
                ;   FINISHES IT.
                ;
00464'054406 FLUSH:     STA     3,R3F
00465'006406 FL1:   JSR@    .GETQE
00466'000402    JMP     FLDON; ERROR RETURN MEANS QUEUE IS EMPTY
00467'000776    JMP     FL1
00470'034402 FLDON:     LDA     3,R3F
00471'001401    JMP     1,3

00472'000000 R3F:   0
00473'000165'.GETQE:    GETQE

                ;
                ;   POINTERS, COMMAND BLOCKS, AND OTHER GOODIES
                ;

                    .ZREL
00000-177777 PMSG:      MESSAGE
00001-000000 CODAD:     0
00002-000000 CPTR:      0
00003-000420 DASTART:DASCON
00004-000521 KBLKADR:KBCON
00005-000000 AS:    0;  TOP OF STACK OF AVAILABLE DCB'S
00006-000000 KQF:       0;  FRONT OF COMMAND QUEUE. 0 MEANS QUEUE IS EMPTY.
00007-000000 KQM:       0;  CURRENT ACTION POINT IN COMMAND QUEUE.
                        ;  USED AND MAINTAINED BY INTERRUPT ROUTINES.
00010-000000 KQR:       0;  REAR OF COMMAND QUEUE.
00011-000000 SERRCNT:       0;  COUNT OF SOFT ERRORS
00012-000000 HERRCNT:       0;  COUNT OF HARD ERRORS
00013-000452 WWLOC:     WWCON
00014-000501 INTVEC:    IVCON


0009  ALTKD
```

```
                .NREL

                ;
                ;   THIS ROUTINE SETS OR INCREMENTS A DISK ADDRESS. IT
                ;   USES A TABLE OF CURRENT, MIN, AND MAX VALUES FOR THE
                ;   VARIOUS FIELDS OF A DISK ADDRESS. ENCODED ADDRESS GOES IN
                ;   CODAD. SKIP IF SUCCESS.
                ;
00474'054464 SETMIN:  STA     3,R3T
00475'030464    LDA     2,TABMIN
00476'024500 MINLP:   LDA     1,TABMAX
00477'132513    SG  1,2; SKIP IF TABMAX>CURTAB
00500'000427    JMP     ENCODE;     DONE
00501'025001    LDA     1,1,2;      GET MIN FOR THIS FIELD
00502'045000    STA     1,0,2;      MAKE IT CURRENT
00503'024474    LDA     1,TABINC
00504'133000    ADD     1,2; MOVE TO NEXT FIELD
00505'000771    JMP     MINLP

00506'054452 INCAD:   STA     3,R3T
00507'030452    LDA     2,TABMIN
00510'024466 INCLP:   LDA     1,TABMAX
00511'132513    SG  1,2; SKIP IF TABMAX>CURTAB
00512'000444    JMP     OFFEND;     CARRY PROPAGATED TO NON-EXISTENT FIELD
00513'025000    LDA     1,0,2;      GET CURRENT VALUE FOR THIS FIELD
00514'125400    INC 1,1; ADD 1
00515'021002    LDA     0,2,2;      GET MAXIMUM VALUE FOR THIS FIELD
00516'122512    SLE 1,0; IF CURRENT<=MAX, UPDATE CURRENT AND QUIT
00517'000403    JMP     CARRY
00520'045000    STA     1,0,2;      UPDATE CURRENT
00521'000406    JMP     ENCODE

00522'025001 CARRY:   LDA     1,1,2;      GET MINIMUM VALUE FOR THIS FIELD
00523'045000    STA     1,0,2;      STORE IT IN CURRENT
00524'024453    LDA     1,TABINC;   MOVE TO NEXT FIELD
00525'133000    ADD     1,2
00526'000762    JMP     INCLP

00527'024433 ENCODE:  LDA     1,SECTOR
00530'125020    MOVZ    1,1; FORCE A ZERO CARRY
00531'125120    MOVZL   1,1
00532'125120    MOVZL   1,1
00533'125120    MOVZL   1,1
00534'125120    MOVZL   1,1
00535'125120    MOVZL   1,1
00536'125120    MOVZL   1,1
00537'125120    MOVZL   1,1
00540'125120    MOVZL   1,1
00541'125120    MOVZL   1,1
00542'030431    LDA     2,CYL
00543'147000    ADD     2,1
00544'125120    MOVZL   1,1
00545'030420    LDA     2,HEAD
00546'147000    ADD     2,1
00547'125120    MOVZL   1,1
00550'030420    LDA     2,DSKNO
00551'147000    ADD     2,1
00552'125120    MOVZL   1,1
0010  ALTKD
```

```
00553'044001-   STA     1,CODAD
00554'034404    LDA     3,R3T
00555'001401    JMP     1,3

00556'034402 OFFEND:  LDA     3,R3T
00557'001400    JMP     0,3


00560'000000 R3T:   0
00561'000562'TABMIN:  SECTOR
00562'000000 SECTOR:  0
00563'000000    0
00564'000013    13
00565'000000 HEAD:    0
00566'000000    0
00567'000001    1
00570'000000 DSKNO:   0
00571'000000    0
00572'000000    0
00573'000000 CYL:   0
00574'000100    100; DON'T WIPE OUT THE BOOT AREA
00575'000312    312
00576'000576'TABMAX:  .
00577'000003 TABINC:  3
00600'000001 C1:    1


                ;
                ;   REGARDLESS OF THE FINAL STATUS, GIVE A SKIP RETURN
                ;
00601'001401 NOERR:   JMP     1,3


                ;
                ;   CHECK THAT THE COMMAND COMPLETED CORRECTLY.
                ;

00602'025001 CKDERR:  LDA     1,STATUS,2;  GET THE FINISHING STATUS
00603'020411    LDA     0,ERMSK
00604'107415    AND#    0,1,SNR
00605'001401    JMP     1,3;  CORRECT RETURN
00606'054405    STA 3 CKDRET
00607'006000-   JSR@ PMSG
00610'001616'   CDER
00611'062000    ERROR
00612'002401    JMP@ CKDRET

00613'000000 CKDRET:  0


00614'000277 ERMSK:   277


                ;
                ;   COMPARE THE RECORD READ WITH THE ONE ORIGINALLY WRITTEN
                ;
00615'054427 CDAT:    STA     3,R3C
00616'004764    JSR CKDERR;   CHECK THE STATUS
0011  ALTKD
```

```
00617'002425     JMP@    R3C;     IF ERROR
00620'050422     STA     2,R2C
00621'021005     LDA     0,HDRWD,2
00622'024421     LDA     1,IHDR1
00623'106414     SUB#    0,1,SZR; CHECK THAT SUBSTITUTION REALLY WORKS
00624'000406     JMP     LNOTEQ
00625'006422     JSR @ARTST;     COMPARE THE RANDOM DATA
00626'000411     JMP     NOTEQ; IF ERROR
00627'030413     LDA     2,R2C;   COMPARE IS OK
00630'034414     LDA     3,R3C
00631'001401     JMP     1,3

00632'061000 LNOTEQ:  DIR
00633'006000-    JSR@ PMSG
00634'001641'    LFAIL
00635'061001     EIR
00636'062000     ERROR
00637'030403 NOTEQ:   LDA     2,R2C
00640'034404     LDA     3,R3C
00641'001400     JMP     0,3

00642'000000 R2C:   0
00643'171717 IHDR1:     171717
00644'000000 R3C:   0
00645'000010 C8:    10
00646'000400 C256:  400
00647'000752'ARTST:     RTST


                ;
                ;    RANDOM DATA GENERATOR -- GENERATES INTO LAB AND DAT
                ;
00650'054447 RGEN:      STA     3,R3RG
00651'050447     STA     2,R2RG
00652'024440     LDA     1,DWDMX
00653'044440     STA     1,DWDCT
00654'031003     LDA     2,LABEL,2;   R2 POINTED TO THE DCB
00655'034444     LDA     3,APRO
00656'024436     LDA     1,PRMX; COPY THE STARTUP PARAMETERS TO THE
00657'044436     STA     1,PRCT; LABEL BLOCK

00660'025400 RGLPO:     LDA     1,0,3
00661'045000     STA     1,0,2
00662'151400     INC 2,2
00663'175400     INC 3,3
00664'014431     DSZ     PRCT
00665'000773     JMP     RGLPO

00666'030432     LDA     2,R2RG
00667'031004     LDA     2,DATA,2;    ADDRESS OF DATA BLOCK
00670'102400     SUB     0,0

00671'024423 RGLP:      LDA     1,PRMX
00672'044423     STA     1,PRCT
00673'034426     LDA     3,APRO

00674'025400 RGLPA:     LDA     1,0,3
00675'123000     ADD     1,0
0012  ALTKD
```

```
00676'041400     STA     0,0,3
00677'041000     STA     0,0,2
00700'014413     DSZ     DWDCT
00701'151401     INC 2,2,SKP
00702'000405     JMP     RGOK
00703'014412     DSZ     PRCT
00704'175401     INC 3,3,SKP
00705'000764     JMP     RGLP
00706'000766     JMP     RGLPA

00707'034410 RGOK:    LDA     3,R3RG
00710'030410     LDA     2,R2RG
00711'001401     JMP     1,3

00712'000400 DWDMX:   400
00713'000000 DWDCT:   0
00714'000010 PRMX:    10
00715'000000 PRCT:    0
00716'000000 RBEGAD:  0
00717'000000 R3RG:    0
00720'000000 R2RG:    0
00721'000722'APRO:    PRO
00722'177777 PRO:   177777
00723'166666      166666
00724'155555      155555
00725'144444      144444
00726'133333      133333
00727'122222      122222
00730'111111      111111
00731'100000      100000


                  ;
                  ;    RANDOM DATA TEST ROUTINE
                  ;
00732'054765 RTST:    STA     3,R3RG
00733'050765      STA     2,R2RG
00734'024756      LDA     1,DWDMX
00735'044756      STA     1,DWDCT
00736'025003      LDA     1,LABEL,2
00737'044757      STA     1,RBEGAD
00740'031004      LDA     2,DATA,2
00741'102400      SUB     0,0

00742'024752 RTLP:    LDA     1,PRMX
00743'044752      STA     1,PRCT
00744'034752      LDA     3,RBEGAD

00745'025400 RTLPA:   LDA     1,0,3
00746'123000      ADD     1,0
00747'041400      STA     0,0,3
00750'025000      LDA     1,0,2
00751'106414      SUB#    0,1,SZR
00752'000410      JMP     RTERR
00753'014740      DSZ     DWDCT
00754'151401      INC 2,2,SKP
00755'000415      JMP     RTOK
00756'014737      DSZ     PRCT
00757'175401      INC 3,3,SKP
0013  ALTKD
```

```
00760'000762    JMP     RTLP
00761'000764    JMP     RTLPA

00762'061000 RTERR:     DIR
00763'006000-   JSR@ PMSG
00764'001664'   DATF
00765'061001    EIR
00766'062000    ERROR
00767'034730    LDA     3,R3RG
00770'030730    LDA     2,R2RG
00771'001400    JMP     0,3

00772'034725 RTOK:      LDA     3,R3RG
00773'030725    LDA     2,R2RG
00774'001401    JMP     1,3


                ;
                ;   THIS ROUTINE SETS UP DISK DATA AND CONTROL BLOCKS

00775'054434 SUDB:      STA     3,R3SUD
00776'020434    LDA 0  NBUFS
00777'100400    NEG     0,0
01000'176400    SUB     3,3;  SET UP ZERO LINK FOR FIRST BLOCK
01001'030432    LDA     2  FIRBUF
01002'055000 SULP:      STA     3,POINTER,2;      SET STACK LINK
01003'034431    LDA     3,HDRDIS
01004'157000    ADD     2,3
01005'055011    STA     3,HEADER,2;       SET HEADER LINK
01006'034427    LDA     3,DCBLEN
01007'157000    ADD     2,3
01010'055003    STA     3,LABEL,2;    SET LABEL LINK
01011'034425    LDA     3,DLLEN
01012'157000    ADD     2,3
01013'055004    STA     3,DATA,2;     SET DATA LINK
01014'034424    LDA     3,CS2;    SUCCESS INTERRUPT
01015'055007    STA     3,SUCCESS,2
01016'034423    LDA     3,CS4;    FAILURE INTERRUPT
01017'055010    STA     3,FAILURE,2
01020'155000    MOV     2,3;  CURRENT BLOCK BECOMES OLD BLOCK
01021'030416    LDA     2,BLKLEN;   LENGTH OF A BLOCK
01022'173000    ADD     3,2;  ADDRESS OF NEW BLOCK
01023'101404    INC 0,0,SZR;  DONE ENOUGH BLOCKS YET?
01024'000756    JMP     SULP;   NO
01025'054005-   STA     3,AS;     SET UP THE AVAILABLE STACK
01026'040006-   STA     0,KQF;  THE DISK QUEUE IS EMPTY
01027'034402    LDA     3,R3SUD
01030'001401    JMP     1,3

01031'000000 R3SUD:     0
01032'000000 NBUFS:     0
01033'000000 FIRBUF:    0
01034'000005 HDRDIS:    HDRWD-POINTER
01035'000015 DCBLEN:    DCBEND-POINTER
01036'000025 DLLEN:     DCBEND-POINTER+10
01037'000425 BLKLEN:    DCBEND-POINTER+10+400
01040'000002 CS2:   2
01041'000004 CS4:   4
0014  ALTKD
```

```
                ;
                ;   SET UP AND ENABLE INTERRUPTS
                ;
01042'054450 SUINT:     STA     3,R3SU
01043'034455    LDA     3,ATRVEC;   INITIALIZE TRAP VECTOR TO
01044'020453    LDA     0,CS40;     JMP@ TRPC
01045'040455    STA     0,ISUCNT
01046'020453    LDA     0,ARETROU
01047'041400 TSULP:     STA     0,0,3
01050'175400    INC 3,3
01051'014451    DSZ     ISUCNT
01052'000775    JMP     TSULP
01053'034004-   LDA     3,KBLKADR
01054'020440    LDA     0,CS10;  TURN ON DISK SECTOR INTERRUPTS
01055'041403    STA     0,3,3
01056'020451    LDA     0,ABADINT
01057'034014-   LDA     3,INTVEC
01060'024435    LDA     1,NBADI
01061'044441    STA     1,ISUCNT
01062'024431    LDA     1,CY1
01063'030433    LDA     2,DBUGAD;  DON'T MESS WITH THE DEBUGGER'S INT
01064'156414 ISULP:     SUB#    2,3,SZR;  SKIP OVER DEBUGGER INTERRUPT
01065'041400    STA     0,0,3;          INITIALLY MARK ALL INTERRUPTS AS BAD
01066'175400    INC 3,3
01067'122400    SUB     1,0;        THIS LETS US FIND OUT WHICH INTERRUPT
01070'014432    DSZ     ISUCNT
01071'000773    JMP     ISULP
01072'034014-   LDA     3,INTVEC;   NOW FILL IN THE GOOD INTERRUPTS
01073'020430    LDA     0,ASIR;     THE SECTOR INTERRUPT ROUTINE
01074'041403    STA     0,3,3
01075'020427    LDA     0,AXER;     TRANSFER ERROR ROUTINE
01076'041402    STA     0,2,3
01077'020426    LDA     0,AXNER;    TRANSFER NO-ERROR ROUTINE
01100'041401    STA     0,1,3

01101'034013-   LDA     3,WWLOC;    ENABLE ALL INTERRUPTS
01102'102400    SUB     0,0
01103'041401    STA     0,1,3;          MASK OFF ALL INTERRUPTS
01104'061001    EIR
01105'041400    STA     0,0,3;          WIPE OUT PENDING INTERRUPTS
01106'020420    LDA     0,ALLINTS
01107'041401    STA     0,1,3;          UNMASK ALL INTERRUPTS

01110'034402    LDA     3,R3SU
01111'001401    JMP     1,3


01112'000000 R3SU:      0
01113'000001 CY1:   1
01114'000010 CS10:  10
01115'000020 NBADI:     20;  NUMBER OF INTERRUPTS
01116'000517 DBUGAD:        IVCON+16;  DEBUGGER'S INTERRUPT
01117'000040 CS40:  40
01120'000530 ATRVEC:    TRPVEC
01121'001466'ARETROU: RETROU
01122'000000 ISUCNT:    0
01123'001263'ASIR:  SIR
01124'001350'AXER:      XER
01125'001267'AXNER:     XNER
0015  ALTKD
```

```
01126'177777 ALLINTS: 177777
01127'001150'ABADINT: BADINT


                ;
                ;   BAD INTERRUPT. THIS SHOULDN'T HAPPEN.
                ;
01130'010522    ISZ  WHY
01131'010521    ISZ  WHY
01132'010520    ISZ  WHY
01133'010517    ISZ  WHY
01134'010516    ISZ  WHY
01135'010515    ISZ  WHY
01136'010514    ISZ  WHY
01137'010513    ISZ  WHY
01140'010512    ISZ  WHY
01141'010511    ISZ  WHY
01142'010510    ISZ  WHY
01143'010507    ISZ  WHY
01144'010506    ISZ  WHY
01145'010505    ISZ  WHY
01146'010504    ISZ  WHY
01147'010503    ISZ  WHY
01150'054574 BADINT:  STA     3,R3XX
01151'006401    JSR@  .+1
01152'001376'   IREE
01153'020477    LDA   0,WHY;  IS THIS A PARITY INTERRUPT?
01154'101014    MOV#  0,0,SZR
01155'000463    JMP   NONPAR

01156'022475 PARITY:  LDA@  0,.DSPL; SAVE THE DISPLAY POINTER
01157'040475    STA   0,ODSPL

01160'102400    SUB   0,0;      TURN OFF THE DISPLAY
01161'042472    STA@  0,.DSPL

01162'020475    LDA   0,BIGNO;     WAIT FOR DISPLAY TO QUIESCE
01163'040475    STA   0,DLY
01164'014474    DSZ   DLY
01165'000777    JMP   .-1

01166'022004-   LDA@  0,KBLKADR;      WAIT FOR THE DISK TO STOP
01167'101014    MOV#  0,0,SZR
01170'000776    JMP   .-2

01171'061000    DIR;            TURN OFF PARITY BIT IN WW
01172'022464    LDA@  0,.WW
01173'024462    LDA   1,PARCON
01174'124000    COM   1,1
01175'123400    AND   1,0
01176'042460    STA@  0,.WW
01177'061001    EIR

01200'152400    SUB   2,2;      SET UP FOR A MEMORY PASS TO LOOK FOR PARITY
01201'050461    STA   2,NFND;          NUMBER OF ERRORS FOUND ON THIS PASS
01202'020457    LDA   0,LIMIT
01203'040455    STA   0,DLY

01204'021000 PARLP:   LDA   0,0,2;          PICK UP A WORD
0016  ALTKD
```

```
01205'101000    MOV 0 0         ;GIVE THE INTERRUPT TIME TO TRY TO HAPPEN
01206'026450    LDA 1 @.WW
01207'125212    MOVR# 1 1 SZC
01210'000416    JMP     FNDIT;  YES .. AC2 POINTS TO THE CULPRIT

01211'151400 PARNX:     INC 2,2;  NO .. TRY NEXT WORD
01212'014446    DSZ     DLY
01213'000771    JMP     PARLP

01214'020440    LDA     0,ODSPL;      START UP THE DISPLAY AGAIN
01215'042436    STA@    0,.DSPL

01216'020444    LDA     0,NFND;         IF AT LEAST ONE ERROR WAS FOUND, IT
01217'101004    MOV     0,0,SZR;  WAS NOT A PHANTOM
01220'000424    JMP     REINT

01221'061000    DIR
01222'006000-   JSR@ PMSG
01223'001701'   NOPE
01224'062000    ERROR;      PARITY ERROR, NO BAD WORD FOUND

01225'000417    JMP     REINT

01226'010434 FNDIT:     ISZ  NFND;  WE FOUND ANOTHER ERROR
01227'061000    DIR
01230'006000-   JSR@ PMSG
01231'001717'   REALP
01232'062000    ERROR;      PARITY ERROR  AC2 POINTS TO BAD WORD, AC0 HAS DATA READ
01233'022423    LDA 0 @.WW      ;TURN OFF THE P.E. BIT IN WW
01234'101200    MOVR 0 0
01235'101120    MOVZL 0 0
01236'042420    STA 0 @.WW

01237'000752    JMP     PARNX; TRY FOR ANOTHER BAD WORD

01240'061000 NONPAR:  DIR
01241'006000-   JSR@ PMSG
01242'001735'   XXI
01243'062000    ERROR;      UNEXPECTED INTERRUPT, NOT PARITY

01244'010406 REINT:     ISZ  WHY;   MAKE SURE WHY IS POSITIVE
01245'014405    DSZ     WHY
01246'000777    JMP     .-1
01247'034475    LDA     3,R3XX
01250'002401    JMP@    .+1
01251'001425'   IRET

01252'000000 WHY:       0
01253'000420 .DSPL:     420
01254'000000 ODSPL:     0
01255'000001 PARCON:    1
01256'000452 .WW:  452
01257'177777 BIGNO:     177777
01260'000000 DLY:  0
01261'137777 LIMIT:     137777
01262'000000 NFND:      0;   NUMBER OF ERRORS FOUND ON THIS PASS

                ;
                ;   SECTOR INTERRUPT ROUTINE. ADD ONE TO SECTOR COUNTER AND
0017  ALTKD
```

```
                ;    RETURN.
                ;
01263'010403 SIR:    ISZ  SECTCNT
01264'101010         MOV# 0,0; NOP
01265'061002         BRI

01266'000000 SECTCNT: 0



                ;
                ;    TRANSFER NO-ERROR ROUTINE
                ;
01267'054455 XNER:       STA      3,R3XX
01270'006401             JSR@     .+1
01271'001376'            IREE
01272'030007-            LDA      2,KQM
01273'021001             LDA      0,STATUS,2;  PICK UP THE DCB'S STATUS
01274'024451             LDA      1,CX7677
01275'123400             AND      1,0
01276'024450             LDA      1,CX7400
01277'106415             SUB#     0,1,SNR
01300'000404             JMP .+4
01301'006000-            JSR@ PMSG
01302'001753'            WRSTAT
01303'062000             ERROR;       ERROR STATUS OR NO STATUS
01304'024443             LDA      1,CX1;   NEW INTERRUPT PROCESSED WORD
01305'021013 XCOM:       LDA      0,INTSDONE,2;    GET OLD INTERRUPT PROCESSED WORD
01306'101015             MOV#     0,0,SNR; SHOULD BE 0
01307'000404             JMP .+4
01310'006000-            JSR@ PMSG
01311'002001'            IAP
01312'062000             ERROR;       ALREADY PROCESSED BY INTERRUPT
01313'045013             STA      1,INTSDONE,2;    STORE NEW VALUE
01314'031000             LDA      2,POINTER,2;     MOVE TO NEXT COMMAND BLOCK
01315'050007-            STA      2,KQM
01316'151015             MOV#     2,2,SNR; CHECK FOR NULL LINK
01317'000422             JMP      XRET
01320'021001             LDA      0,STATUS,2;  GET THE STATUS OF THE OPERATION
01321'101015             MOV#     0,0,SNR
01322'000417             JMP      XRET;    NOT YET FINISHED
01323'024422             LDA      1,CX7677;    ENSURE THAT THE INTERRUPT HAPPENS
01324'123400             AND      1,0
01325'024421             LDA      1,CX7400
01326'106414             SUB#     0,1,SZR
01327'000403             JMP      XER1
01330'021007             LDA      0,SUCCESS,2;     GET NO-ERROR INTERRUPT WORD
01331'000402             JMP      XSHR
01332'021010 XER1:       LDA      0,FAILURE,2;     GET ERROR INTERRRUPT WORD
01333'026013-XSHR:       LDA@     1,WWLOC;     OR THE INTERRUPT WORD INTO WW
01334'111000             MOV      0,2
01335'133400             AND      1,2
01336'146400             SUB      2,1
01337'123000             ADD      1,0
01340'042013-            STA@     0,WWLOC
01341'034403 XRET:       LDA      3,R3XX
01342'002401             JMP@     .+1
01343'001425'            IRET

0018  ALTKD
```

```
01344'000000 R3XX:      0
01345'007677 CX7677:    7677
01346'007400 CX7400:    7400
01347'000001 CX1:   1


                ;
                ;   TRANSFER ERROR INTERRUPT ROUTINE
                ;
01350'054774 XER:   STA     3,R3XX
01351'006401        JSR@    .+1
01352'001376'       IREE
01353'030007-       LDA     2,KQM
01354'021001        LDA     0,STATUS,2;  GET THIS DCB'S STATUS
01355'024770        LDA     1,CX7677
01356'123400        AND     1,0
01357'024767        LDA     1,CX7400
01360'106414        SUB#    0,1,SZR
01361'000404        JMP .+4
01362'006000-       JSR@ PMSG
01363'002022'       NESE
01364'062000        ERROR;      STATUS IS NO ERROR
01365'123400        AND     1,0
01366'122415        SUB#    1,0,SNR
01367'000404        JMP .+4
01370'006000-       JSR@ PMSG
01371'002050'       NSSE
01372'062000        ERROR;      NO STATUS STORED
01373'024402        LDA     1,CX2
01374'000711        JMP     XCOM

01375'000002 CX2:   2


                ;
                ;   REENABLE DEBUGGER INTERRUPT AND SAVE STATE
                ;
01376'040421 IREE:      STA     0,R0X
01377'044421        STA     1,R1X
01400'050421        STA     2,R2X
01401'022414        LDA 0 @TPCP     ;SAVE TRAP PC IN CASE WE WERE INTERRUPTED OUT OF A
TRAP
01402'040414        STA 0 OTPC
01403'030013-       LDA     2,WWLOC
01404'021001        LDA     0,1,2;          SAVE CURRENT INTERRUPT MASK
01405'040415        STA     0,CMASK
01406'020415        LDA     0,DBMSK
01407'041001        STA     0,1,2;          SET NEW MASK FOR ONLY DEBUGGER
01410'030014-       LDA     2,INTVEC
01411'021377        LDA     0,-1,2;         SAVE INTERRUPT OLD PC
01412'040412        STA     0,OLDPC
01413'061001        EIR
01414'001401        JMP     1,3


01415'000527 TPCP:      527
01416'000000 OTPC:      0
01417'000000 R0X:   0
0019  ALTKD
```

```
01420'000000 R1X:   0
01421'000000 R2X:   0
01422'000000 CMASK:   0
01423'040000 DBMSK:   40000;    THE DEBUGGER INTERRUPT ONLY
01424'000000 OLDPC:   0
                ;
                ;   REENABLE ALL INTERRUPTS AND RETURN CONTROL TO INTERRUPTED
STUFF
                ;
01425'061000 IRET:      DIR
01426'020770    LDA 0 OTPC
01427'042766    STA 0 @TPCP
01430'030013-   LDA     2,WWLOC
01431'020771    LDA     0,CMASK
01432'041001    STA     0,1,2;          RESTORE ORIGINAL MASK
01433'030014-   LDA     2,INTVEC
01434'020770    LDA     0,OLDPC
01435'041377    STA     0,-1,2;         RESTORE OLD PC
01436'020761    LDA     0,R0X
01437'024761    LDA     1,R1X;          RESTORE REGISTERS
01440'030761    LDA     2,R2X
01441'061002    BRI



                ;
                ;   THIS ROUTINE VERIFIES THAT AN INTERRUPT WAS PROCESSED FOR
                ;   THIS DCB AND THAT ITS TYPE WAS CONSISTENT WITH THE
                ;   STATUS ENTERED IN THE DCB.
                ;
01442'054423 CKINT:     STA     3,R3CKS
01443'021001    LDA     0,STATUS,2;  GET THE DCB'S STATUS WORD
01444'024701    LDA     1,CX7677
01445'123400    AND     1,0
01446'024700    LDA     1,CX7400
01447'106414    SUB#    0,1,SZR
01450'000413    JMP     CKIER
01451'024676    LDA     1,CX1;   PROPER INTERRUPT TYPE FOR NO ERROR
01452'021013 CKINCOM: LDA     0,INTSDONE,2;    ACTUAL TYPE
01453'106415    SUB#    0,1,SNR
01454'000405    JMP .+5
01455'061000    DIR
01456'006000-   JSR@ PMSG
01457'002073'   WINT
01460'062000    ERROR;      WRONG TYPE OR NO TYPE
01461'034404    LDA     3,R3CKS
01462'001401    JMP     1,3

01463'024712 CKIER:     LDA     1,CX2;   PROPER INTERRUPT TYPE FOR ERROR
01464'000766    JMP     CKINCOM


01465'000000 R3CKS:     0



                ;
                ;   RETURN FROM TRAP ROUTINE
0020  ALTKD
```

```
                ;
01466'054407 RETROU:  STA     3,R3RU
01467'036407    LDA@  3,ATRPC
01470'054407    STA   3,LOCPC
01471'002401    JMP@  .+1;JUMP TO BREAKPOINTED INSTRUCTION
01472'000001'   BREAK

01473'034402 RR2:  LDA    3,R3RU
01474'002403    JMP@  LOCPC

01475'000000 R3RU:    0
01476'000527 ATRPC:   TRPC
01477'000000 LOCPC:   0


                ;SIZE AND PARCEL OUT MEMORY.  IF THERE IS SPACE BETWEEN 1000
                ;AND THE BOTTOM OF THE PROGRAM, WE USE IT FOR DISK BUFFERS
01500'054453 SIZEM:     STA 3 SIZRET
01501'030455    LDA 2 EOM  ;SIZE THE MEMORY
01502'126520    SUBZL 1 1
01503'045001 SIZEL:     STA 1 1 2
01504'021001    LDA 0 1 2
01505'132400    SUB 1 2
01506'122404    SUB 1 0 SZR
01507'000774    JMP SIZEL
01510'052447    STA 2 @PLIM
01511'020443    LDA 0 SOMEM    ;START OF MEMORY
01512'024443    LDA 1 SOP   ;START OF PROGRAM
01513'106414    SUB# 0 1 SZR
01514'000403    JMP SIZED
                ;OTHERWISE, WE USE SPACE FROM THE END OF THE DEBUGGER TO THE END
OF
                ;CORE
01515'145000    MOV 2 1
01516'020442    LDA 0 EOP
01517'042442 SIZED:     STA 0 @PBUFS    ;AC0 CONTAINS THE START OF THE BUFFERS,AC1 THE
END
01520'042442    STA 0 @DBMS
01521'036442    LDA@ 3 BUFSZ
01522'152401    SUB 2 2 SKP
01523'151400 BUFL:      INC 2 2
01524'163000    ADD 3 0
01525'106032    ADCZ# 0 1 SZC  ;SKGE
01526'000775    JMP BUFL
01527'052435    STA 2 @NBP      ;AC2 = NUMBER OF BUFFERS WHICH WILL FIT IN THE GIVEN
AREA

                ;NOW WE GO THROUGH MEMORY AND CORRECT PARITY
01530'061001    EIR
01531'102400    SUB 0 0
01532'042420    STA 0 @UGHX    ;CLEAR NWW
01533'036424    LDA 3 @PLIM
01534'152520    SUBZL 2 2
01535'021400 PQLP:      LDA 0 0 3
01536'041400    STA 0 0 3
01537'025400    LDA 1 0 3
01540'106414    SUB# 0 1 SZR
01541'000401    JMP .+1
01542'026410    LDA 1 @UGHX   ;CHECK FOR ERROR
01543'125213    MOVR# 1 1 SNC
01544'000402    JMP .+2
01545'000401    JMP .+1      ;BREAK HERE TO GET THE BAD NEWS
 0021  ALTKD
```

```
01546'156404     SUB 2 3 SZR
01547'000766     JMP PQLP
01550'034403     LDA 3 SIZRET
01551'001401     JMP 1 3

01552'000452 UGHX:     452
01553'000000 SIZRET:   0
01554'001000 SOMEM:    1000
01555'000000'SOP:      INIT
01556'176777 EOM:      176777
01557'001261'PLIM:     LIMIT    ;TOP OF MEMORY FOR PARITY SCAN
01560'177777 EOI       PHIAD
01561'001033'PBUFS:    FIRBUF
01562'000010'DBMS:     DBMP
01563'001037'BUFSZ:    BLKLEN
01564'001032'NBP:      NBUFS

                ;
                ;    CLEAR THE STATUS AND INTERRUPTS DONE FIELDS
                ;    IF THE LABEL BLOCK IS TO BE CHECKED, CLEAR IT FIRST
                ;

01565'054424 SETLAB:   STA      3,R3SET

01566'102400     SUB      0,0
01567'041001     STA      0,STATUS,2;  CLEAR DCB'S STATUS
01570'041013     STA      0,INTSDONE,2;     AND INTERRUPTS PROCESSED

01571'021002     LDA      0,DSKCOMM,2
01572'024420     LDA      1,CKLM
01573'123400     AND      1,0
01574'024417     LDA      1,CKLV
01575'106414     SUB#     0,1,SZR
01576'000411     JMP      ESETLAB;     NOT CHECKING THE LABEL

01577'020415     LDA      0,SL10;   LENGTH OF THE LABEL BLOCK
01600'040415     STA      0,SLCNT

01601'035003     LDA      3,LABEL,2
01602'102400     SUB      0,0

01603'041400 SETLLP:   STA      0,0,3;      CLEAR A WORD
01604'175400     INC 3,3
01605'014410     DSZ      SLCNT
01606'000775     JMP      SETLLP

01607'034402 ESETLAB: LDA      3,R3SET
01610'001401     JMP      1,3

01611'000000 R3SET:    0
01612'000060 CKLM:     60;  MASKS ONLY LABEL PART OF COMMAND
01613'000020 CKLV:     20;  VALUE OF LABEL FIELD IF CHECKING
01614'000010 SL10:   10;  LENGTH OF LABEL BLOCK
01615'000000 SLCNT:    0


                CDER:  .TXT     /     CONTROLLER REPORTED BAD STATUS<15>/
01616'020040
0022  ALTKD
```

```
01617'020040
01620'020103
01621'047516
01622'052122
01623'047514
01624'046105
01625'051040
01626'051105
01627'050117
01630'051124
01631'042504
01632'020102
01633'040504
01634'020123
01635'052101
01636'052125
01637'051415
01640'000000
                LFAIL:  .TXT    /       LABEL BLOCK SUBSTITUTION FAILED<15>/
01641'020040
01642'020040
01643'020114
01644'040502
01645'042514
01646'020102
01647'046117
01650'041513
01651'020123
01652'052502
01653'051524
01654'044524
01655'052524
01656'044517
01657'047040
01660'043101
01661'044514
01662'042504
01663'006400
                DATF:   .TXT    /       DATA COMPARE FAILED<15>/
01664'020040
01665'020040
01666'020104
01667'040524
01670'040440
01671'041517
01672'046520
01673'040522
01674'042440
01675'043101
01676'044514
01677'042504
01700'006400
                NOPE:   .TXT    /       PHANTOM PARITY ERROR<15>/
01701'020040
01702'020040
01703'020120
01704'044101
01705'047124
01706'047515
0023  ALTKD
```

```
01707'020120
01710'040522
01711'044524
01712'054440
01713'042522
01714'051117
01715'051015
01716'000000
                REALP: .TXT     /       PARITY ERROR DETECTED<15>/
01717'020040
01720'020040
01721'020120
01722'040522
01723'044524
01724'054440
01725'042522
01726'051117
01727'051040
01730'042105
01731'052105
01732'041524
01733'042504
01734'006400
                XXI:    .TXT    /       UNEXPECTED INTERRUPT<15>/
01735'020040
01736'020040
01737'020125
01740'047105
01741'054120
01742'042503
01743'052105
01744'042040
01745'044516
01746'052105
01747'051122
01750'052520
01751'052015
01752'000000
                WRSTAT:     .TXT    /       NO-ERROR INTERRUPT GAVE WRONG STATUS<15>/
01753'020040
01754'020040
01755'020116
01756'047455
01757'042522
01760'051117
01761'051040
01762'044516
01763'052105
01764'051122
01765'052520
01766'052040
01767'043501
01770'053105
01771'020127
01772'051117
01773'047107
01774'020123
01775'052101
01776'052125
0024  ALTKD
```

```
01777'051415
02000'000000
                IAP:    .TXT    /       INTERRUPT ALREADY PROCESSED<15>/
02001'020040
02002'020040
02003'020111
02004'047124
02005'042522
02006'051125
02007'050124
02010'020101
02011'046122
02012'042501
02013'042131
02014'020120
02015'051117
02016'041505
02017'051523
02020'042504
02021'006400
                NESE:   .TXT    /       NO-ERROR STATUS FROM ERROR INTERRUPT<15>/
02022'020040
02023'020040
02024'020116
02025'047455
02026'042522
02027'051117
02030'051040
02031'051524
02032'040524
02033'052523
02034'020106
02035'051117
02036'046440
02037'042522
02040'051117
02041'051040
02042'044516
02043'052105
02044'051122
02045'052520
02046'052015
02047'000000
                NSSE:   .TXT    /       NO STATUS FROM ERROR INTERRUPT<15>/
02050'020040
02051'020040
02052'020116
02053'047440
02054'051524
02055'040524
02056'052523
02057'020106
02060'051117
02061'046440
02062'042522
02063'051117
02064'051040
02065'044516
02066'052105
0025  ALTKD
```

```
02067'051122
02070'052520
02071'052015
02072'000000
                WINT:   .TXT    /       WRONG INTERRUPT TYPE FOR ERROR<15>/
02073'020040
02074'020040
02075'020127
02076'051117
02077'047107
02100'020111
02101'047124
02102'042522
02103'051125
02104'050124
02105'020124
02106'054520
02107'042440
02110'043117
02111'051040
02112'042522
02113'051117
02114'051015
02115'000000
                HARDERR: .TXT   /       SOFT ERROR HAS HARDENED<15>/
02116'020040
02117'020040
02120'020123
02121'047506
02122'052040
02123'042522
02124'051117
02125'051040
02126'044101
02127'051440
02130'044101
02131'051104
02132'042516
02133'042504
02134'006400
                NEWER:  .TXT    /       NEW ERROR STATUS NOTICED<15>/
02135'020040
02136'020040
02137'020116
02140'042527
02141'020105
02142'051122
02143'047522
02144'020123
02145'052101
02146'052125
02147'051440
02150'047117
02151'052111
02152'041505
02153'042015
02154'000000
                .END
0026  ALTKD
```

```
DBMSK  001423'
DBUGA  001116'
DCBEN  000015
DCBLE  001035'
DIR    061000
DLLEN  001036'
DLY    001260'
DODCB  000114'
DSKAD  000006
DSKBI  000353'
DSKCO  000002
DSKNO  000570'
DWDCT  000713'
DWDMX      000712'
EFND   000253'
EIR 061001
ENCOD  000527'
EOM    001556'
EOP    001560'
EQCOM  000440'
EQDON  000461'
EQUE   000414'
ERMSK  000614'
ERROR  062000
ESETL  001607'
EXTRA  000456'
FAILU  000010
FINIS  000012
FIRBU  001033'
FIREL  000441'
FL1 000465'
FLDON  000470'
FLUSH  000464'
FNDIT  001226'
GC1    000151'
GCB    000150'
GETQE  000165'
GQ1    000166'
HARDE  002116'
HARDT  000351'
HARDY  000256'
HDRDI  001034'
HDRWD      000005
HEAD   000565'
HEADE  000011
HERRC  000012-
IAD    000112'
IAP 002001'
IDONE  000102'
IHDR   000107'
IHDR1  000643'
IMORE  000035'
INCAD  000506'
INCLP  000510'
INIT   000000'
INITA  000013'
INTSD  000013
INTVE  000014-
IREE   001376'
  0028  ALTKD
```

```
IRET    001425'
ISUCN   001122'
ISULP   001064'
IVCON   000501
KBCON   000521
KBLKA   000004-
KQF     000006-
KQM     000007-
KQR     000010-
LABEL   000003
LFAIL   001641'
LIMIT   001261'
LNOTE   000632'
LOCPC   001477'
LOOK    000240'
MESSA   000000-X
MINLP   000476'
MKDCB   000124'
MKDPA   000147'
MONE    000355'
NBADI   001115'
NBP     001564'
NBUFS   001032'
NESE    002022'
NEWER   002135'
NEND    001262'
NOERR   000601'
NOKCO   000350'
NONPA   001240'
NOPAD   000110'
NOPE    001701'
NOTEM   000436'
NOTEQ   000637'
NSSE    002050'
ODSPL   001254'
OFFEN   000556'
OKCON   000347'
OLDPC   001424'
OTPC    001416'
PARCO   001255'
PARIT   001156'
PARLP   001204'
PARNX   001211'
PBUFS   001561'
PHIAD   001560'X
PLIM    001557'
PMSG    000000-
POINT   000000
PQLP    001535'
PRCT    000715'
PRMX    000714'
PRO     000722'
PULLI   000331'
R0X     001417'
R1X     001420'
R2C     000642'
R2RG    000720'
R2X     001421'
R2ZAP   000346'
  0029 ALTKD
```

R3C 000644'
R3CKS 001465'
R3DOD 000123'
R3F 000472'
R3G 000164'
R3MKD 000146'
R3Q 000344'
R3RG 000717'
R3RU 001475'
R3SET 001611'
R3SL 000345'
R3SU 001112'
R3SUD 001031'
R3T 000560'
R3XX 001344'
RBEGA 000716'
RDBLK 000072'
RDIT 000062'
RDMOR 000064'
REALP 001717'
REDO 000031'
REINT 001244'
REMOV 000335'
RES 000204'
RESTA 000326'
RESTH 000352'
RETRO 001466'
RGEN 000650'
RGLP 000671'
RGLPA 000674'
RGLPO 000660'
RGOK 000707'
RKADR 000413'
RR2 001473'
RSNBO 000356'
RSNEN 000377'
RSNTO 000400'
RSTBI 000354'
RTERR 000762'
RTLP 000742'
RTLPA 000745'
RTOK 000772'
RTST 000732'
SDBAD 000053'
SECTC 001266'
SECTO 000562'
SERRC 000011-
SERRS 000014
SETLA 001565'
SETLL 001603'
SETMI 000474'
SIR 001263'
SIZED 001517'
SIZEL 001503'
SIZEM 001500'
SIZRE 001553'
SL10 001614'
SLCNT 001615'
SMAD 000111'
0030 ALTKD

```
SMASH  000104'
SOFT   000272'
SOMEM 001554'
SOP 001555'
SRCHL  000215'
STATU  000001
STKEM  000161'
SUBST  000101'
SUCCE  000007
SUDB   000775'
SUINT  001042'
SULP   001002'
TABIN  000577'
TABMA  000576'
TABMI  000561'
TPCP   001415'
TRPC   000527
TRPVE  000530
TSULP  001047'
UGHX   001552'
VALID  000463'
WHY    001252'
WINT   002073'
WRSTA  001753'
WRTAL  000106'
WRTBL  000054'
WRTIT  000033'
WWCON      000452
WWLOC      000013-
XCOM   001305'
XER    001350'
XER1   001332'
XNER   001267'
XRET   001341'
XSHR   001333'
XXI 001735'
ZAPLP  000320'
.DSPL  001253'
.GETQ  000473'
.RBOT  000050'
.REST  000401'
.RR2   000012'
.RTOP  000051'
.WW    001256'
```