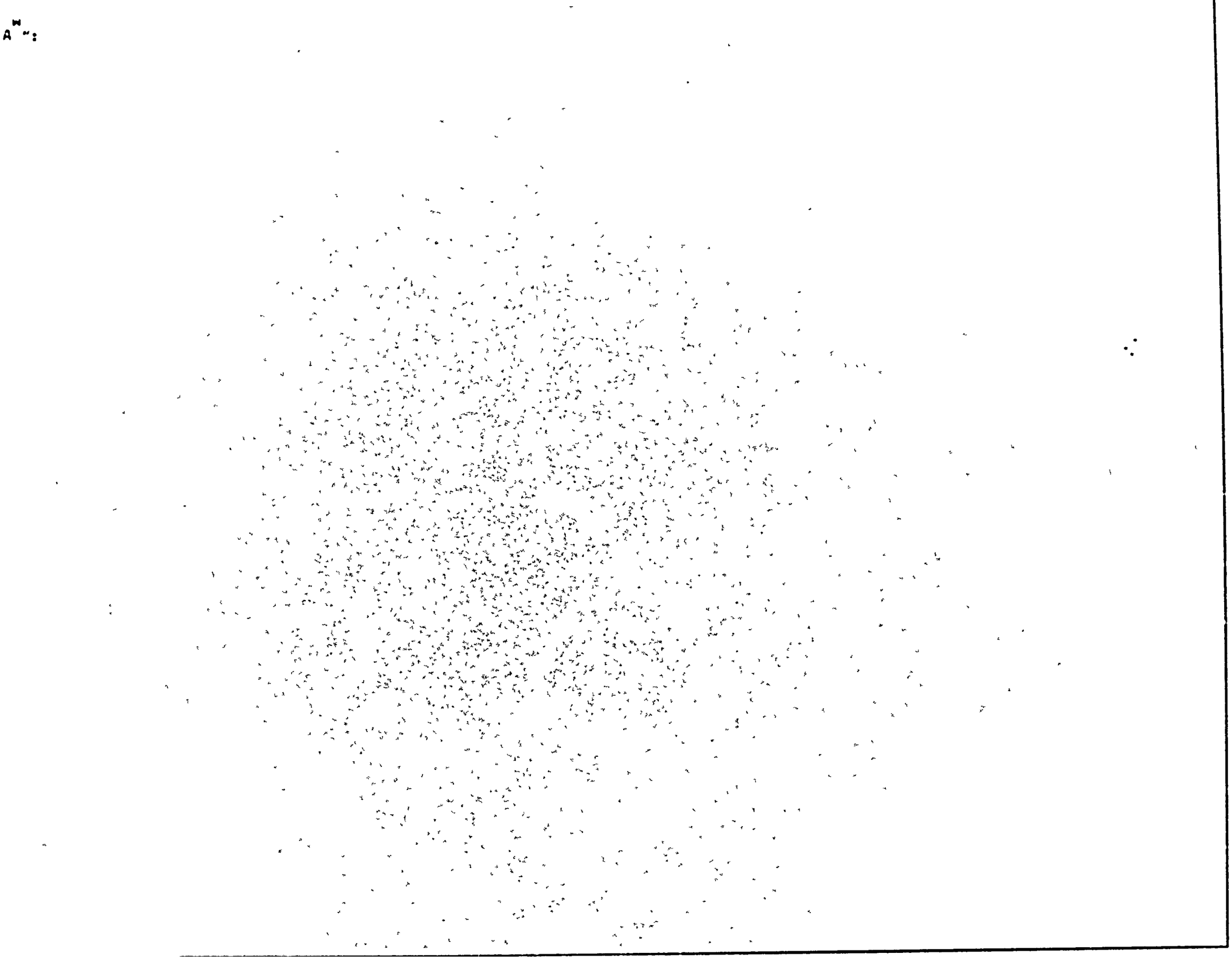
Micro Fiche Scan

Name of device(s) tested:
    RA60/80/81,UDA50A,KDA50Q
Test description:
    UDA50/KDA50 DECX MOD
MAINDEC Number or Package Identifier (after SEP 1977):
    CXDUBD0
Fiche Document Part Number:
    AH-S915D-MC
Fiche preparation date unknown, using copyright year:
    1984
Image resolution:
    1-bit black&white, compressed for minimal file size



IDENTIFICATION

PRODUCT CODE: AC S914D MC

PRODUCT NAME: CXDUBDO UDA50A/KDA50 Q DECx MOD

PRODUCT DATE: 20-SEP-1984

MAINTAINER: ROGER OAKEY

AUTHOR: JOHN MERTZ

```
.ENABL LC

.REM &
```

TABLE OF CONTENTS

1.0 ABSTRACT
2.0 REQUIREMENTS
3.0 START UP
4.0 PASS DEFINITION
5.0 EXECUTION TIME
6.0 CONFIGURATION REQUIREMENTS
7.0 DEVICE/OPTION SETUP
8.0 MODULE OPERATION
9.0 OPERATION OPTIONS
10.0 PRINTOUTS
11.0 DUAL PORT OPERATION
12.0 GLOSSARY
13.0 BIBLIOGRAPHY

1.0 ABSTRACT

The exerciser will be similar to that of other disk subsys tem exercisers. Writes will be performed to the disks fol- lowed by read and compare of the data read. The controller will do all error retrying. Errors will be reported on the console TTY.

All desired disk drives on the controller will be exercised simultaneously. If disk accessing is not required, then data written will go only as far as the controller's RAM memory.

If the results of the exerciser requires more information, two other PDP-11 diagnostic programs are available. They are:

CZUDHA0 - UDA50A/KDA50-Q and Disk Drive Diagnostic
CZUDIA0 - UDA50A/KDA50-Q Disk Drive Exerciser
CZUDKA0 UDA50A/KDA50 Q Disk Formatter.

2.0 REQUIREMENTS

Hardware for all cases:
One DEC/X11 module configures for one UDA50A or KDA50-Q

controller.

Hardware for disk accessing:
One controller with at least one drive is the minimum amount or one controller with four drives is the maximum amount.

Hardware for no disk accessing:
One controller is the only requirement.

Storage: DUBD requires

Decimal words -- 4096 MAX

3.0 START-UP

On the initial start, the program will clear bit1 of 'SR1' and type the following messages.

```
DUBD0 PA:0060162      APC: 000674     PASS 000000

'IF YOU WISH TO DESTROY CUSTOMER DATA, SET BIT1 (NOT BIT0)
 IN SWITCH REGISTER 1(SR1) OF DUBD? EQUAL TO 1.'

DUBD0 PA:0060210      APC: 000722     PASS 000000
'! OPERATING WITH NO DISK ACCESSING !'
```

This will occur reguardless of the condition of SR1 (bit1) at configure time.

If the operator wishes to exercise the drive, SR1 (bit1) must be modified at location 16 of CXDUBD0 module (see section 9). This can be accomplished by using the 'MOD' command supplied by the DECX11 run time system. Unless the program is reloaded or the operator modifies the location again, the contents of SR1 will remain the same on all subsequent starts.

On all subsequent starts, the condition of SR1 (bit1) will type to terminal in the following manor.

If bit1 of SR1 is equal to 0 (zero), the following warning will be typed.

```
DUBD0 PA:0060210      APC: 000722     PASS 000000
'! OPERATING WITH NO DISK ACCESSING !'
```

If bit1 of SR1 is equal to 1 (one), the following warning will be typed.

```
DUBD0 PA:0060210      APC: 000722     PASS 000000
'! CUSTOMER DATA WILL BE DESTROYED !'
```

<<<< NOTE >>>>
When this DEC/X11 module runs in diskless mode, its data rate exceeds all other devices. This may cause erroneous data lates from other devices.

4.0 PASS DEFINITION

One pass of the DUBD module consists of 512 iterations of the basic test sequence (write, read, data-check). The test sequence writes a user defined number of words (default is 256) words, reads 256 words, and data-compare same.

5.0 EXECUTION TIME

The default execution time of one pass of DUBD running alone on a PDP-11/44 under sequential disk accessing mode will be approximately 20 seconds. Under random accessing mode, the time is 40 seconds. For no disk accessing, the time is five seconds

6.0 CONFIGURATION REQUIREMENTS

Default Parameters:

DEVADR: 172150, VECTOR: 154, BR1: 4, DEVCNT: 1, SR1: 0, SR2: 0

REQUIRED PARAMETERS:

Additional controller module(s) configured must have different Unibus address(es) and vector(s).

7.0 DEVICE/OPTION SETUP

For disk mode, make certain that all units are powered up, write enabled, connected to a controller via the SDI and ready.

For diskless mode, make certain the controller is powered up.

8.0 MODULE OPERATION

TEST SEQUENCE DISK MODE:

A. Setup device register addresses and module variables. Set controller characteristic.
B. Reset all units on line and drop all that are not.

C. Get a unit address.
D. Get a disk address and a fresh block of data.
E. Do a write - if errors, report.
F. Do a read -- if errors, report.
G. Do a data-check -- if errors, report and continue.
H. Make unit available.
I. Wait for available attention message.
J. If end of pass, report and go to D.
K. If end of testing unit, go to C; else go to D.

Blocks determined defective won't be replaced by the exerciser during this sequence. The exerciser makes full use of the controller which does revectoring on its own.

TEST SEQUENCE DISKLESS MODE:

A. Get a fresh block of data.
B. Do a write to controller RAM buffer -- if errors, report.
C. Do a read from controller RAM buffer -- if errors,report.
D. Do a data-check -- if errors, report and continue.
E. If end of pass, report.
F. Go to A.

9.0 OPERATION OPTIONS

One or more software switch registers can be used by the module program general purpose switches. These words are used to define or specify a unique device option or to point to a specific routine in the module. Any option must be specified by the operator before the module is run. Switch Register 1 has the following characteristics.

SR1 Bit 1 set (1): Allow disk transfers.
<<<< NOTE >>>> IF SET, CUSTOMER DATA WILL BE DESTROYED!
reset (0): No disk transfers.

SR1 Bit 2 set (1): Do not report errors as they occur.
reset (0): Report errors as they occur.

SR1 Bit 3 set (1): Do not print error summary at end of pass.
reset (0): Print error summary at end of pass.

SR1 Bit 9 set (1): Run Dual port mode (only valid if SR1 Bit 1 is set)
reset (0): Do not run Dual port mode

SR1 Bit 10 set (1): Select random block addressing.(only valid if SR1 Bit 1 is set)
reset (0): Select sequential block

addressing.

SR1 Bit 11 set (1): Bypass data compare.
reset (0): Do data comapare.

Switch register 2 has the following characteristics.

SR2 Bits 0 to 5: Burst rate.

A burst rate to speed up NPR transfers by the controller can be used. This value is 6 bits maximum and set up in SR2 at configure time.

<<<< NOTE >>>>
The DVID1 mask reflects the number of units chosen for testing and which units on the system are to be tested. Example: If DVID1 contains a 1, only the first unit found on the system will be tested. A unit's order on the system is judged by its unit number. The lowest unit number zero (0). Unit 0 would be the first tested on the system.

If DVID1 contains a 10, the fourth unit on the system will be tested. If the first two units are chosen, DVID1 is 3. Four consecutive units means DVID1 is 17. Six units, DVID1 is 77.

If there is not a unit corresponding to the DVID1 bit setting, the bit set in DVID1 gets cleared. The exerciser will readjust the mask and drop the nonexistent units if more units are chosen than actually are present. The module is dropped if all DVID1 bits are cleared.

If the number of units chosen is less then the actual number of units present, only the desired units will be used during the exercise.

<<<< ANOTHER NOTE >>>>
Make sure all subunited drives are accounted for. Destroying customer data is not desireble.

<<<< ONE MORE NOTE >>>>
If SR1 Bit 3 is reset, a summary status is printed every 15 passes. This status is formatted as follows:

```
DUBD0 PA: 00060470  ACP: 001210  PASS #00000

SOFT ERROR COUNT #00000        *** HARD ERROR COUNT #00000
CHECK DATA ERROR COUNT #00000
```

10.0 PRINTOUTS

A. Most printouts have the standard formats described in the DEC/X11 document.

B. Non-standard printouts include error messages which dump the following:

1) Summary status
2) Flags and endcode
3) Unit number
4) Byte count
5) Hi 16-bit LBN value
6) Lo 16-bit LBN value
7) Extended address
8) Physical address

All values except for PASS, RUNTIME and ERRCNT are printed in octal. PASS, RUNTIME and ERRCNT are printed in decimal. Example:

```
DUBD0 PA: 00064116 APC: 004630 PASS: 00000 ERRCNT: 00001
CSRA: 172150 CSRC: 000000 ASTAT: 000006 ERRTYP: 000006
RUNTIME: 000:00:22

DUBD0 PA: 00064052 APC: 004564 PASS: 00000

STATUS ENDCOD UNITNU BYTECO HI LBN LO LBN EXTADR PHYADR
000006 000242 000005 000000 000003 116321 000001 062100
```

STATUS - responce of the command sent to the controller. This is contained in the last five bits of the word. Here is a list of status codes.

```
 0 -    success
 1 -    invalid command
 2 -    command aborted
 3 -    unit offline
 4 -    unit available
 5 -    media error
 6 -    write protected
 7      compare error
10 -    data error
11 -    host buffer access error
12 -    controller error
13 -    drive error
```

ENDCOD ending code of the command sent. This shows what command was sent to the UDA. Here is a list of all possible endcodes this module uses.

```
100 -   AVAILABLE ATTENTION MESSAGE (not a command but a
              message sent to the host from the UDA)
200 -   INVALID COMMAND
203     GET UNIT STATUS
204 -   SET CONTROLLER CHARACTERISTICS
210 -   AVAILABLE
211     ONLINE
```

230 - MAINTENANCE READ
231 - MAINTENANCE WRITE
241 - READ
242 - WRITE

UNITNU - unit number of the drive that is being accessed. This is not relevent if the user is running diskless mode.

BYTECO - size of the buffer in bytes.

HI LBN - high logical block number (upper 16 bits) which tells the user where on the disk the data is going. This is only valid for disk mode.

LO LBN - low logical block number (lower 16 bits).

EXTADR - extended address of the read/write buffer.

PHYADR - physical address of the read/write buffer.

C. If the controller failed to pass its internal diagnostic, one of the following messages will be printed.

If the diagnostic found a fault:

```
DUBDO PA: 00062052 APC: 002564 PASS: 00000
CONTROLLER INIT ERROR, FOUND BY DIAGNOSTIC
SA REGISTER = xxxxxx IN STEP yyyyy
ADDR = zzzzzz
```

If a step bit was not set as expected during the initialization sequence of the controller:

```
DUBDO PA: 00062152 APC: 002664 PASS: 00000
CONTROLLER INIT ERROR, STEP NOT SET
SA REGISTER = xxxxxx IN STEP yyyyy
ADDR = zzzzzz
```

If data passed back from the controller was not equal to the expected value:

```
DUBDO PA: 00062252 APC: 002764 PASS: 00000
CONTROLLER INIT ERROR, EXPECTED DATA WAS INCORRECT
SA REGISTER = xxxxxx IN STEP yyyyy
ADDR = zzzzzz
```

Where xxxxxx can have any of the following values and meanings:

```
104000 - Fatal sequncer error
104040 - D processor ALU error
104041 - D proc ROM parity error
105102 - D proc with no Board #2 or RAM parity error
105105 - D proc RAM buffer error
105152 - D proc SDI error
105153 - D proc write mode wrap SERDES error
```

105154 - D proc read mode SERDES, RSGEN, and ECC error
106040 - U proc ALU error
106041 - U proc Control Register error
106042 - U proc DFAIL/ROM parity error/Board #1 test count is wrong
106047 - U proc Constant ROM error with D proc running SDI test
106055 - Unexpectant trap found, aborted diagnostic
106071 - U proc ROM error
106072 - U proc ROM parity error
106200 - Step 1 data error (MSB not set)
107103 - U proc RAM parity error
107107 - U proc RAM buffer error
107115 - Board #2 test count was wrong
112300 - Step 2 error
122240 - NPR error
122300 - Step 3 error
142300 - Step 4 error

Where yyyyy is the step in which the error was found.

Where zzzzzz is the address of the UDA.

If the maximmum number of retries has been exceeded, the following message will be printed.

```
DUBD0 PA: 00061414 APC:002126 PASS #00000

RETRY COUNT EXCEEDED, ABORT
```

This means the controller did not successfully complete the initialization in four passes. The module is then dropped.

D. If the UDA did not successfully clear the ring buffer in the host area, the following message will be printed.

```
DUBD0 PA: 00061414 APC:002126 PASS #00000

RING AREA NOT CLEARED
```

This is a fatal error. It means that the controller did not access host memory that the cpntroller would use to communicate with the host. The module is then dropped.

E. If the SAregister displays a non-zero value after the initialization sequence is done, the followng message will be printed.

```
DUBD0 PA: 00064252 APC: 004764 PASS: 00000
SA REGISTER IS NOT ZERO, = xxxxxx
CONTROLLER IS GOING THROUGH INITIALIZATION
```

Where xxxxxx can have the following values and meanings.

004400 - controller has been inited by either a bus init or by writing irto the IP register.
100001 - bus envelope/packet read error (parity or timeout)

100002 - bus envelopepacket write error (parity or timeout)
100003 - controller ROM and RAM parity error
100004 - controller RAM parity error
100005 - controller ROM parity error
100006 - bus ring read error
100007 - bus ring write error
100010 - bus interrupt master failure
100011 - Host access timeout error
100012 - Host exceeded credit limit
100013 - controller SDI hardware fatal error
100014 - DM XFC fatal error
100015 - Hardware timeout of instruction loop
100016 - Invalid virtual circuit identifier
100017 - Interrupt write error on bus

E. If a drive is dropped by the exerciser, one of the following messages will be printed.

If the drive had an error it could not handle properly after an iteration, the following message will be printed:

```
DUBD0 PA: 00063012 APC: 003524 PASS #00000

DRIVE 00000 DROPPED.
DEVICE ID BIT = 000001
ERRORS CAUSED DRIVE TO BE DROPPED
```

If the drive was not found by the exerciser, the following message will be printed:

```
DUBD0 PA: 00063012 APC: 003524 PASS #00000

DRIVE 00000 DROPPED.
DEVICE ID BIT = 000001
UNIT WAS NOT FOUND BY THE EXERCISER
```

If there were more device count bits set than the actual number of drives found, the following message will be printed:

```
DUBD0 PA: 00063012 APC: 003524 PASS #00000

DRIVE 00000 DROPPED.
DEVICE ID BIT = 000001
DVIC% BIT SET HIGHER THAN ACTUAL # OF DRIVES FOUND
```

Solution: try a lesser number of units in DVID1 (loc 14)

11.0 DUAL PORT OPERATION

To run a dual port operation, set bit9 of SR1. The exerciser will check the unit to see if it is offline or available.

The controller will retain control of a unit until the MSCP Available command is entered by the host. During this time, the other controller is not allowed access to the unit through the other port between the write and read. The other controller senses when the unit becomes available and takes it. The MSCP Available command is only executed if SR1 bit 9 and SR1 bit 1 are set. This allows dual porting and disk accessing respectively.

DEC/X11 will only dual port a drive with another DEC/X11 exerciser.

12.0 GLOSSARY

DUBD follows the module name format described in the DEC/X11 Programmer's Guide.

DU-- Identifies the hardware and thus the module.

--B- Distiguishes between two or more different modules for the same generic device. The sequence A, B, C, ETC. must be used for each additional example.

---D Specifies the module revision.

IOMODX is a type of module in an extended input/output mode. These modules are interrupt driven and are capable of input/output operation. Some added capabilities provided include:

- Use of monitor supplied write buffers.
- Ability to change the size of the write buffers.
- Access to the monitor's check data utility.
- Conversion routines to get 18 bit addresses from 16 bit addresses.

13.0 BIBLIOGRAPHY

CXQUAA0 'DEC/X11 USER'S MANUAL' Sept 1978

CXQAFD0 'DEC/X11 PROGRAMMERS'S GUIDE' Sept 1978

CXQUBA0 'DEC/X11 CROSS-REFERENCE MANUAL' Sept 1978

```
        2                                       .DSABL LC
        3
```

```
                                        .SBTTL  MODULE HEADER BLOCK
                                                IOMODX <DUBD  >,172150,154,4,0,0,1000,104,RBUF,256.,,256.
000000                                          MODULE  150000,DUBD  ,172150,154,4,0,0,1000,104,RBUF,256.,,256.
000000                                          .TITLE  DUBD DEC/X11 SYSTEM EXERCISER MODULE
                                        ;       DDXCOM  VERSION  6      23-MAY-78
                                                .LIST   BIN
                                        ;*****************************************************************************
                                        BEGIN:
000000                                  MODNAM: .ASCII  /DUBD / ;MODULE NAME.
000000    104     125     102
000003    104     040
000005    000                           XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006  172150                          ADDR:   172150+0                ;1ST DEVICE ADDR.
000010  000154                          VECTOR: 154+0                   ;1ST DEVICE VECTOR.
000012    200                           BR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013    000                           BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014  000001                          DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016  000000                          SR1:    OPEN                    ;SWITCH REGISTER 1
000020  000000                          SR2:    OPEN                    ;SWITCH REGISTER 2
000022  000000                          SR3:    OPEN                    ;SWITCH REGISTER 3
000024  000000                          SR4:    OPEN                    ;SWITCH REGISTER 4
                                        ;*****************************************************************************
000026  150000                          STAT:   150000                  ;STATUS WORD.
000030  000660'                         INIT:   START                   ;MODULE START ADDR.
000032  000252'                         SPOINT: MODSP                   ;MODULE STACK POINTER.
000034  000000                          PASCNT: 0                       ;PASS COUNTER.
000036  001000                          ICONT:  1000                    ;# OF ITERATIONS PER PASS=1000
000040  000000                          ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042  000000                          SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044  000000                          HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046  000000                          SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050  000000                          HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052  000000                          SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054  000000                          RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056                                  CONFIG:                         ;RESERVED FOR MONITOR USE
000056  000000                          RES1:   0                       ;RESERVED FOR MONITOR USE
000060  000000                          RES2:   0                       ;RESERVED FOR MONITOR USE
000062  000000                          SVR0:   OPEN                    ;LOC TO SAVE R0.
000064  000000                          SVR1:   OPEN                    ;LOC TO SAVE R1.
000066  000000                          SVR2:   OPEN                    ;LOC TO SAVE R2.
000070  000000                          SVR3:   OPEN                    ;LOC TO SAVE R3.
000072  000000                          SVR4:   OPEN                    ;LOC TO SAVE R4.
000074  000000                          SVR5:   OPEN                    ;LOC TO SAVE R5.
000076  000000                          SVR6:   OPEN                    ;LOC TO SAVE R6.
000100  000000                          CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102                                  SBADR:                          ;ADDR OF GOOD DATA, OR
000102  000000                          ACSR:   OPEN                    ;CONTENTS OF CSR.
000104                                  WASADR:                         ;ADDR OF BAD DATA, OR
000104  000000                          ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106                                  ERRTYP:                         ;TYPE OF ERROR
000106  000000                          ASB:    OPEN                    ;EXPECTED DATA.
000110  000000                          AWAS:   OPEN                    ;ACTUAL DATA.
000112  001036'                         RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114  000000                          WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116  000000                          WDF:    OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120  000000                          INT:    OPEN                    ;# OF INTERRUPTS PER ITERATION
000122  000104                          IDNUM:  104                     ;MODULE IDENTIFICATION NUMBER=104
```

```
                                MODULE HEADER BLOCK

000124  007126          RBUFVA: RBUF                    ;READ BUFFER VIRTUAL ADDRESS
000126  000000          RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
000130  000000          RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132  000400          RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
000134  000000          WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136  000000          WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140  000400          WBUFRQ: 256.                    ;WRITE BUFFER SIZE REQUESTED
000142  000000          WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144  000000          CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146  000000          CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150  000000          FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                  .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                .NLIST
                                .WORD   0
                                .LIST
                                .ENDR
000252                  MODSP:
                        ;*****************************************************************
                        ;*****************************************************************
                        .SBTTL  MODULE STORAGE AREA
                        ;       VERSION 1.0     FOR RELEASE
                        ;       VERSION 1.1     NO LONGER TEST AFTER STEP 4
                        ;                       NO LONGER WAIT FOR INTERRUPT AFTER SENDING MSCP AVAILABLE COMMAND
                        ;       VERSION 2.0     USE BIT 9 IN SR1 FOR DUAL PORTING. (DON'T SEND MSCP AVAILABLE
                        ;                       COMMAND IF WE WANT JUST SEQUENTIAL OR RANDOM ACCESS MODE --
                        ;                       IN OTHER WORDS, ONLY SEND ONLINE COMMAND ONCE DURING PASS UNLESS
                        ;                       DUAL PORT MODE)
                        ;       VERSION 3.0     KDA50-Q SUPPORT ADDED

        000002          SR.XFR = BIT01  ;NO DISK TRANSFER 0 = NO DISK TRANSFER, 1 = DO DISK TRANSFER
        000004          SR.REP = BIT02  ;REPORT ERROR AS THEY OCCUR 0 = REPORT, 1 = DON'T REPORT
        000010          SR.SUM = BIT03  ;REPORT ERRORS ON END OF PASS 0 = REPORT, 1 = DON'T REPORT
        001000          SR.DUA = BIT09  ;DUAL PORT 0 = NO DUAL PORT, 1 = DUAL PORT
        002000          SR.SEQ = BIT10  ;RANDOM (NOT SEQUENTIAL) DISK ADDRESSING 0 = SEQUENTIAL, 1 = RANDOM
        004000          SR.CMP = BIT11  ;NO DATA COMPARE 0 = DO DATA COMPARE, 1 = DON'T DO DATA COMPARE

                        ;IPREG: .WORD   0               ; CONTROLLER POLLING REG
000252  000000          SAREG:  .WORD   0               ; CONTROLLER STATUS REG

000254  000000          CINTR:  .WORD   0               ;COMMAND INTERRUPT INDICATOR
000256  000000          RINTR:  .WORD   0               ;RESPONCE INTERRUPT INDICATOR

000260                  RSPONC: .BLKW   2.              ;MESSAGE RING
000264                  COMMND: .BLKW   2.              ;COMMAND RING

000270  000000          CMDREF: .WORD   0               ;COMMAND REFERENCE NUMBER

000272  000000          RSPLEN: .WORD   0               ;RESPONCE PACKET LENGTH
000274  000000          RSPVIR: .WORD   0               ;RESPONCE PACKET VIRTUAL CIRCUIT
000276                  RSPACK: .BLKW   24.             ;RESPONCE PACKET

000356  000000          CMPLEN: .WORD   0               ;COMMAND PACKET LENGTH
000360  000000          CMPVIR: .WORD   0               ;COMMAND PACKET VIRTUAL CIRCUIT
000362                  CMPACK: .BLKW   24.             ;COMMAND PACKET

000442  000264'         VA:     .WORD   COMMND          ;GENERIC VIRTUAL ADDRESS FOR GETPA
000444  000000          PA:     .WORD   OPEN            ;GENERIC PHYSICAL ADDRESS
```

```
MODULE STORAGE AREA

                                EA:     .WORD   OPEN            ;GENERIC EXTENDED ADDRESS

                                RBFFEA: .WORD   0       ;READ  BUFFER EXTENDED ADDRESS SAVE AREA
                                WBFFEA: .WORD   0       ;WRITE BUFFER EXTENDED ADDRESS SAVE AREA

                                NUM:    .WORD   0               ;ADDRESS USED IN OTOA
                                OLDPA:  .WORD   0               ;THE OLD PHYSICAL ADDRESS
                                OLDEA:  .WORD   0               ;THE OLD EXTENDED ADDRESS TO CHECK IF
                                                                ;       CONTROLLER WILL BE REINITED

                        000017  PRTNUM = 15.                    ;PRINT MESSAGE EVERY 15TH TIME
                                PRNMSG: .WORD   PRTNUM          ;PRINT WORD SAVES THE VALUE TO CHECK FOR WHEN
                                                                ; THE NEXT TIME AN END OF PASS MESSAGE IS WRITTEN
                        002260  TIMER = 1200.                   ;TIMER VALUE TO WAIT 2-3 SECONDS AFTER DAP COMMAND

                                EXPAV:  .WORD   177777          ;EXPECTING AN AVAILABLE ATTENTION MESSAGE = 0
                                                                ;NOT EXPECTING AN AVAILABLE ATTENTION MESSAGE = 177777

                                ADR1:   .BLKB   6
                                        .BYTE   0
                                ADR2:   .BLKB   6
                                        .BYTE   0
                                ADR3:   .BLKB   6
                                        .BYTE   0
                                ADR4:   .BLKB   6
                                        .BYTE   0
                                ADR5:   .BLKB   6
                                        .BYTE   0
                                ADR6:   .BLKB   6
                                        .BYTE   0
                                ADR7:   .BLKB   6
                                        .BYTE   0
                                ADR8:   .BLKB   6
                                        .BYTE   0
                                        .EVEN
```

```
                                .SBTTL MORE MODULE STORAGE
                                ;
                                ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS
                                ;NEEDED FOR MAP 22 ROUTINE
000556  000000                  PA18:   .WORD   0
000560  000000                  XMEM:   .WORD   0
000562  000000                  PA22:   .WORD   0
000564  000000                  EA22:   .WORD   0

000566  000000                  SECL:   .WORD   0               ;CURRENT SECTOR LO ORDER ADDRESS
000570  000000                  SECH:   .WORD   0               ;CURRENT SECTOR HI ORDER ADDRESS

000572  000000                  UNSZL:  .WORD   0               ;UNIT SIZE LO ORDER LIMIT FROM ONLINE CMND
000574  000000                  UNSZH:  .WORD   0               ;UNIT SIZE HI ORDER LIMIT

000576  003300                  LIMIT:  .WORD   3300            ;4K - 1200 = MOST WORDS MAITW CAN TAKE

000600  000001                  DVICE:  .WORD   1               ;DEVICE TO TEST
000602  000000                  UNITNO: .WORD   0               ;UNIT NUMBER
000604  000000                  TRY:    .WORD   0               ;NUMBER OF TRIES
000606  000001                  PORTID: .WORD   1               ;BIT POSITION SELECTS THE PORT
000610  000000                  UNITFL: .WORD   0               ;SAVE UNIT FLAGS
000612  000000                  WORK:   .WORD   0               ;TEMPORARY WORK AREA

        005670                  TIMOUT = 3000.                  ;TIME OUT GADGE
        000004                  RLIM = 4                        ;RETRY LIMIT

000614  000000  000001          TABLEW: .WORD   0,1             ;TABLE ENTRY UNITNO,PORTID
000620  177777  177777                  .WORD   -1,-1           ;CURRENT LAST TABLE ENTRY
000624                                  .BLKW   12.             ;REST OF TABLE
000654  177777  177777          TEND:   .WORD   -1,-1           ;END MARKER

                                ;S:     .WORD   0,0,0,0,0,0,0,0         ;FOR HARD AND SOFT ERRORS
                                ;       .WORD   177777

                                ;TABLE: .WORD   S               ;EACH ENTRY OF THE TABLE POINTS TO
                                ;       .WORD   S+2             ;THE CORRESPONDING ENTRY OF S.
                                ;       .WORD   S+4             ;THIS IS USED IN HRDER & SOFER
                                ;       .WORD   S+6
                                ;       .WORD   S+10
                                ;       .WORD   S+12
                                ;       .WORD   S+14
                                ;       .WORD   S+16
                                ;       .WORD   177777
```

```
                        .SBTTL  MODULE PRIVATE DATA

000001                  BIT00 = 1
000002                  BIT01 = 2
000004                  BIT02 = 4
000010                  BIT03 = 10
000020                  BIT04 = 20
000040                  BIT05 = 40
000100                  BIT06 = 100
000200                  BIT07 = 200
000400                  BIT08 = 400
001000                  BIT09 = 1000
002000                  BIT10 = 2000
004000                  BIT11 = 4000
010000                  BIT12 = 10000
020000                  BIT13 = 20000
040000                  BIT14 = 40000
100000                  BIT15 = 100000

                        ;
                        ;       ERROR BITS
                        ;
000000                  ERR.0 = 0       ;NOT DEFINED
000001                  ERR.1 = 1       ;DATA ERROR
000003                  ERR.3 = 3       ;CONTROLLER NOT READY
000006                  ERR.6 = 6       ;DRIVE NOT READY, OFF LINE OR NON EXESTENT
000032                  ERR.32 = 32     ;NPR ERROR
```

```
                                .SBTTL CONTROLLER BIT DEFINITIONS
                                
                                ; SA REGISTER UNIVERSAL READ BITS
                                
004000                          SA.S1= 004000                   ;STEP 1 STATUS BIT
010000                          SA.S2= 010000                   ;STEP 2 STATUS BIT
020000                          SA.S3= 020000                   ;STEP 3 STATUS BIT
040000                          SA.S4= 040000                   ;STEP 4 STATUS BIT
100000                          SA.ERR= 100000                  ;ERROR INDICATOR
                                
                                ; SA REGISTER ERROR STATUS BITS
                                
003777                          SA.ERC= 003777                  ;ERROR CODE
                                
                                ; SA REGISTER STEP ONE READ BITS
                                
                                ;;SA.CTP=       003400                  ;CONTROLLER TYPE
002000                          SA.NSI= 002000                  ; NON SETTABLE INTERRUPT
001000                          SA.Q22= 001000                  ; 22 BIT ADDRESS BUS
000400                          SA.DIA= 000400                  ; DIAG BIT IN SA REGISTER
000100                          SA.MAP= 000100                  ; MAPPING BIT
000040                          SA.SM = 000040                  ; SPECIAL MODE BIT FOR KDA50-Q
                                
                                ; SA REGISTER STEP ONE WRITE BITS
                                
000177                          SA.VEC= 000177                  ; INTERRUPT VECTOR (DIVIDED BY 4)
000200                          SA.INT= 000200                  ; INTERRUPT ENABLE DURING INITIALIZATION
003400                          SA.RSP= 003400                  ; MESSAGE RING LENGTH
034000                          SA.CMD= 034000                  ; COMMAND RING LENGTH
                                
                                ; SA REGISTER STEP TWO READ BITS
                                
000177                          SA.VCE= 000177                  ;INTERRUPT VECTOR ECHO
000200                          SA.INE= 000200                  ;INTERRUPT ENABLE ECHO
                                
                                ; SA REGISTER STEP TWO WRITE BITS
                                
                                                                ;LOW ORDER MESSAGE RING BYTE ADDRESS
000001                          SA.PRG= 000001                  ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
                                
                                ; SA REGISTER STEP THREE READ BITS
                                
000017                          SA.RSE= 000017                  ;RESPONCE RING LENGTH ECHO
000360                          SA.CME= 000360                  ;COMMAND RING LENGTH ECHO
                                
                                ; SA REGISTER STEP THREE WRITE BITS
                                
                                                                ;HIGH ORDER MESSAGE RING BYTE ADDRESS
040000                          SA.LFC= 040000                  ;LAST FAILURE CODE REQUEST
                                
                                ; SA REGISTER STEP FOUR READ BITS
                                
000377                          SA.MCV= 000377                  ; CONTROLLER MICROCODE VERSION
                                
                                ; SA REGISTER STEP FOUR WRITE BITS
                                
000001                          SA.GO=  BIT0                    ;GO BIT TO START CONTROLLER FIRMWARE
```

```
                                .SBTTL COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS
                        
                        
100000                  RG.OWN= BIT15                   ;SET WHEN CONTROLLER OWNS RING
040000                  RG.FLG= BIT14                   ;FLAG BIT
                        
                                ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
                        
000010                  HC.SIZ= 8.                      ;SIZE OF HOST COMM AREA IN BYTES
000060                  PKTSIZ= 48.                     ;SIZE OF PACKETS IN BYTES
                        
000000                  HC.RES= 0.                      ;RESPONCE RING START
000002                  HC.RCT= 2.                      ;RESPONCE RING CONTROL WORD
000004                  HC.CMD= 4.                      ;COMMAND RING START
000006                  HC.CCT= 6.                      ;CONTROL RING CONTROL WORD
000276'                 HC.RPK= RSPACK                  ;START OF RESPONCE PACKET BUFFER
000356'                 HC.CPK= HC.RPK+PKTSIZ           ;START OF COMMAND PACKET BUFFER
```

COMMAND PACKET OPCODES

```
                                .SBTTL COMMAND PACKET OPCODES

                                OP.ABO= 01                      ;ABORT COMMAND
                                OP.ACC= 20                      ;ACCESS COMMAND
                                OP.AVL= 10                      ;AVAILABLE COMMAND
                                OP.CCD= 21                      ;COMPARE CONTROLLER DATA COMMAND
                                OP.CMP= 40                      ;COMPARE HOST DATA COMMAND
                                OP.DAP= 13                      ;DETERMINE ACCESS PATHS COMMAND
                                OP.ERS= 22                      ;ERASE COMMAND
                                OP.FLU= 23                      ;FLUSH COMMAND
                                OP.GCS= 02                      ;GET COMMAND STATUS COMMAND
                                OP.GUS= 03                      ;GET UNIT STATUS COMMAND
                                OP.ONL= 11                      ;ONLINE COMMAND
                                OP.RD=  41                      ;READ COMMAND
                                OP.RPL= 24                      ;REPLACE COMMAND
                                OP.SCC= 04                      ;SET CONTROLLER CHARACTERISTICS COMMAND
                                OP.SUC= 12                      ;SET UNIT CHARACTERISTICS COMMAND
                                OP.WR=  42                      ;WRITE COMMAND
                                OP.MRD= 30                      ;MAINTENANCE READ COMMAND
                                OP.MWR= 31                      ;MAINTENANCE WRITE COMMAND
                                OP.END= 200                     ;END PACKET FLAG
                                OP.AVA= 100                     ;AVAILABLE ATTENTION MESSAGE
                                OP.ERL= 101                     ;ERROR LOG ATTENTION MESSAGE
                                OP.SHC= 102                     ;SHADOW COPY COMPLETE ATTENION MESSAGE
                                OP.ACP= 102                     ;ACCESS PATH ATTENTION MESSAGE

                                ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
                                ;PACKET FLAG TO THE COMMAND OPCODE. THE UNKNOWN COMMAND END PACKET CONTAINS
                                ;JUST THE END PACKET FLAG IN ITS OPCODE FIELD.
```

```
                        .SBTTL COMMAND MODIFIERS

                        
040000                  MD.CMP= 040000          ;COMPARE
100000                  MD.EXP= 100000          ;EXPRESS REQUEST
010000                  MD.ERR= 010000          ;FORCE ERROR
004000                  MD.SCH= 004000          ;SUPPRESS CACHING (HIGH SPEED)
002000                  MD.SCL= 002000          ;SUPPRESS CACHING (LOW SPEED)
001000                  MD.SEC= 001000          ;SUPPRESS ERROR CORRECTION
000400                  MD.SER= 000400          ;SUPPRESS ERROR RECOVERY
000200                  MD.SSH= 000200          ;SUPPRESS SHADOWING
000100                  MD.WBN= 000100          ;WRITE-BACK (NON-VOLATILE)
000040                  MD.WBV= 000040          ;WRITE BACK (VOLATILE)
000001                  MD.SPD= 000001          ;SPIN-DOWN
000001                  MD.FEU= 000001          ;FLUSH ENTIRE UNIT
000002                  MD.VOL= 000002          ;VOLATILE ONLY
000001                  MD.NXU= 000001          ;NEXT UNIT

                        .SBTTL END PACKET FLAGS

                        
000200                  EF.BBR= 000200          ;BAD BLOCK REPORTED
000100                  EF.BBU= 000100          ;BAD BLOCK UNREPORTED
000040                  EF.LOG= 000040          ;ERROR LOG GENERATED
000020                  EF.SEX= 000020          ;SERIOUS EXCEPTION


                        .SBTTL UNIT FLAGS

                        
000001                  UF.CMR= 000001          ;COMPARE READS
000002                  UF.CMW= 000002          ;COMPARE WRITES
010000                  UF.RPL= 010000          ;HOST INITIATED BAD BLOCK REPLACEMENT
040000                  UF.INA= 040000          ;INACTIVE SHADOW SET UNIT
000200                  UF.RMV= 000200          ;REMOVEABLE MEDIA
004000                  UF.SCH= 004000          ;SUPPRESS CACHING (HIGH SPEED)
002000                  UF.SCL= 002000          ;SUPPRESS CACHING (LOW SPEED)
000040                  UF.WBN= 000040          ;WRITE-BACK (NON-VOLATILE)
020000                  UF.WPH= 020000          ;WRITE PROTECT(HARDWARE)
010000                  UF.WPS= 010000          ;WRITE PROTECT(SOFTWARE OR VOLUME)
000004                  UF.576= 000004          ;576 BYTE SECTORS
```

```
                                .SBTTL CONTROLLER FLAGS

                        000200  CF.AVL= 000200                  ;ENABLE AVAILABLE ATTENTION MESSAGES
                        000100  CF.MSC= 000100                  ;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
                        000040  CF.OTH= 000040                  ;ENABLE OTHER HOST'S ERROR LOG MESSAGES
                        000020  CF.THS= 000020                  ;ENABLE THIS HOST'S ERROR LOG MESSAGES
                        000002  CF.SHD= 000002                  ;SHADOWING
                        000001  CF.576= 000001                  ;576 BYTE SECTORS

                                .SBTTL COMMAND PACKET OFFSETS

                                ;                       GENERIC COMMAND PACKET OFFSETS:
                        000000  P.CRF=   0.                    ;COMMAND REFERENCE NUMBER
                        000004  P.UNIT=  4.                    ;UNIT NUMBER
                        000010  P.OPCD=  8.                    ;OPCODE
                        000012  P.MOD=  10.                    ;MODIFIERS
                        000014  P.BCNT= 12.                    ;BYTE COUNT
                        000020  P.BUFF= 16.                    ;BUFFER DESCRIPTOR
                        000020  P.ADPA= 16.                    ;BUFFER'S PHYSICAL ADDRESS (P.BUFF)
                        000022  P.ADEA= 18.                    ;BUFFER'S EXTENDED ADDRESS (P.BUFF+2)
                        000034  P.LBN=  28.                    ;LOGICAL BLOCK NUMBER
                        000040  P.SFTW= 32.                    ;SOFTWARE WORDS

                                ;                       ABORT AND GET COMMAND STATUS COMMAND PACKET OFFSETS:
                        000014  P.OTRF= 12.                    ;OUTSTANDING REFERENCE NUMBER

                                ;                       ONLINE AND SET UNIT CHARACTERISTICS COMMAND PACKET OFFSETS:
                        000016  P.UNFL= 14.                    ;UNIT FLAGS
                        000020  P.HSTI= 16.                    ;HOST IDENTIFIER
                        000024  P.UNTI= 20.                    ;UNIT IDENTIFIER
                        000034  P.ELGF= 28.                    ;ERROR LOG FLAGS
                        000040  P.SHUN= 32.                    ;SHADOW UNIT
                        000042  P.CPSP= 34.                    ;COPY SPEED

                                ;                       REPLACE COMMAND PACKET OFFSETS:
                        000014  P.RBN=  12.                    ;REPLACEMENT BLOCK NUMBER

                                ;                       SET CONTROLLER CHARACTERISTICS COMMAND PACKET OFFSETS:
                        000014  P.VRSN= 12.                    ;MSCP VERSION
                        000016  P.CNTF= 14.                    ;CONTROLLER FLAGS
                        000020  P.HTMO= 16.                    ;HOST TIMEOUT
                        000022  P.USEF= 18.                    ;USE FRACTION
                        000024  P.TIME= 20.                    ;QUAD-WORD TIME AND DATE

                                ;                       MAINTENANCE READ AND MAINTENANCE WRITE COMMAND PACKET OFFSETS:
                        000034  P.RGID= 28.                    ;REGION ID
                        000040  P.RGOF= 32.                    ;REGION OFFSET
```

```
                        .SBTTL END PACKET OFFSETS

                        ;               GENERIC END PACKET OFFSETS:
000000                  P.CRF=   0.                     ;COMMAND REFERENCE NUMBER
000004                  P.UNIT=  4.                     ;UNIT NUMBER
000010                  P.OPCD=  8.                     ;OPCODE (ALSO CALLED ENDCODE)
000011                  P.FLGS=  9.                     ;END PACKET FLAGS
000012                  P.STS=  10.                     ;MODIFIERS
000014                  P.BCNT= 12.                     ;BYTE COUNT
000034                  P.FBBK= 28.                     ;FIRST BAD BLOCK
000040                  P.SFTW= 32.                     ;SOFTWARE WORDS

                        ;               GET COMMAND STATUS END PACKET OFFSETS:
000014                  P.OTRF= 12.                     ;OUTSTANDING REFERENCE NUMBER
000020                  P.CMST= 16.                     ;COMMAND STATUS

                        ;               GET UNIT STATUS END PACKET OFFSETS:
000014                  P.MLUN= 12.                     ;MULTI-UNIT CODE
000016                  P.UNFL= 14.                     ;UNIT FLAGS
000020                  P.HSTI= 16.                     ;HOST IDENTIFIER
000024                  P.UNTI= 20.                     ;UNIT IDENTIFIER
000040                  P.SHUN= 32.                     ;SHADOW UNIT
000042                  P.SHST= 34.                     ;SHADOW STATUS
000044                  P.TRCK= 36.                     ;TRACK SIZE
000046                  P.GRP=  38.                     ;GROUP SIZE
000050                  P.CYL=  40.                     ;CYLINDER SIZE
000054                  P.RCTS= 44.                     ;RCT TABLE SIZE
000056                  P.RBNS= 46.                     ;RBN / TRACK
000057                  P.RCTC= 47.                     ;RCT COPIES

                        ;               ONLINE AND SET UNIT CHARACTERISTICS
000014                  P.MLUN= 12.                     ;MULTI-UNIT CODE
000016                  P.UNFL= 14.                     ;UNIT FLAGS
000020                  P.HSTI= 16.                     ;HOST IDENTIFIER
000024                  P.UNTI= 20.                     ;UNIT IDENTIFIER
000040                  P.SHUN= 32.                     ;SHADOW UNIT
000044                  P.UNSZ= 36.                     ;UNIT SIZE
000050                  P.VSER= 40.                     ;VOLUME SERIAL NUMBER

                        ;               SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS:
000014                  P.VRSN= 12.                     ;MSCP VERSION
000016                  P.CNTF= 14.                     ;CONTROLLER FLAGS
000020                  P.CTMO= 16.                     ;CONTROLLER TIMEOUT
000022                  P.CNCL= 18.                     ;CONTROLLER COMMAND LIMIT
000024                  P.CNTI= 20.                     ;CONTROLLER ID
000034                  P.MEDI= 28.                     ;MEDIA TYPE
000042                  P.SHST= 34.                     ;SHADOW STATUS

                        ;ERROR LOG ATTENTION MESSAGE PACKET OFFSETS

000000                  P.CRF=   0.                     ;COMMAND REFERENCE NUMBER
000004                  P.UNIT=  4.                     ;UNIT NUMBER
000006                  P.CNT=   6.                     ;COUNT
000010                  P.OPCD=  8.                     ;OPCODE
000011                  P.FLGS=  9.                     ;ERROR LOG FLAGS
000012                  P.SZOF= 10.                     ;SIZE OR OFFSET
000014                  P.LGDT= 12.                     ;START OF ERROR LOG DATA
```

```
                        .SBTTL ERROR LOG FLAGS

000200                  EF.FRS= 000200                  ;FIRST PACKET
000100                  EF.LST= 000100                  ;LAST PACKET
000001                  EF.MIS= 000001                  ;MESSAGE MISSING

                        ;ERROR LOG MESSAGE OFFSETS

000000                  L.EVNT=  0.                     ;EVENT CODE
000002                  L.SLOT=  2.                     ;SLOT NUMBER
000004                  L.CNTI=  4.                     ;CONTROLLER IDENTIFIER
000014                  L.CNTI= 12.                     ;CONTROLLER SOFTWARE REVISION
000015                  L.CHVR= 13.                     ;CONTROLLER HARDWARE REVISION
000016                  L.UNTI= 14.                     ;UNIT IDENTIFIER
000026                  L.USVR= 22.                     ;UNIT SOFTWARE REVISION
000027                  L.UHVR= 23.                     ;UNIT HARDWARE REVISION
000030                  L.ERLC= 24.                     ;ERROR LOCATION
000034                  L.CYL=  28.                     ;CYLINDER
000040                  L.GRP=  32.                     ;GROUP
000041                  L.TRCK= 33.                     ;TRACK
000042                  L.SCTR= 34.                     ;SECTOR
000044                  L.VSER= 36.                     ;VOLUME SERIAL NUMBER
000050                  L.DATA= 40.                     ;EVENT DEPENDENT DATA

                        ;STATUS AND EVENT COE DEFINITIONS

000037                  ST.MSK= 37                      ;STATUS / EVENT CODE MASK
000040                  ST.SUB= 40                      ;SUB-CODE MULTIPLIER
000000                  ST.SUC=  0                      ;SUCCESS
000001                  ST.CMD=  1                      ;INVALID COMMAND
000002                  ST.ABO=  2                      ;COMMAND ABORTED
000003                  ST.OFL=  3                      ;UNIT-OFFLINE
000004                  ST.AVL=  4                      ;UNIT-AVAILABLE
000005                  ST.MFE=  5                      ;MEDIA ERROR
000006                  ST.WPR=  6                      ;WRITE PROTECTED
000007                  ST.CMP=  7                      ;COMPARE ERROR
000010                  ST.DAT= 10                      ;DATA ERROR
000011                  ST.HST= 11                      ;HOST BUFFER ACCESS ERROR
000012                  ST.CNT= 12                      ;CONTROLLER ERROR
000013                  ST.DRV= 13                      ;DRIVE ERROR
000037                  ST.DIA= 37                      ;MESSAGE FROM AN INTERNAL DIAGNOSTIC

                        ;
                        ;       SUBCODES FOR ST.OFL
                        ;
000040                  SC.NVL =  40                    ;NO VOLUME MOUNTED
                                                        ; OR DRIVE DISAVLED VIA RUN/STOP SWITCH
000100                  SC.IOP = 100                    ;UNIT INOPERATIVE
000400                  SC.DIS = 400                    ;UNIT DISABLED BY FIELD SERVICE
                                                        ; OR INTERNAL DIAGNOSTIC
000200                  SC.DUP = 200                    ;DUPLICATE UNIT NUMBER
                        ;
                        ;       SUBCODES FOR ST.DRV
                        ;
000040                  SC.STO =  40                    ;SDI RESPONCE TIME OUT
000100                  SC.INV = 100                    ;INVALID SDI RESPONCE
```

```
                        .SBTTL  MODULE CODE
                        ;******************************************************************************
                        ;                                                                             :
                        ;       INIT VALUES                                                           :
                        ;       INIT CONTROLLER                                                       :
                        ;       XFER TO DISK?                                                         :
                        ;               F FOR J = 1,CYCLE LIMIT                                       :
                        ;                 MAITENANCE WRITE                                            :
                        ;                 MAITENANCE READ                                             :
                        ;                 CHECK DATA?                                                 :
                        ;                       T CHECK                                               :
                        ;               NEXT J                                                        :
                        ;               T FOR J = 1,CYCLE LIMIT                                       :
                        ;                 GET UNIT STATUS                                             :
                        ;                       IF DRIVE IS NOT AVAILABLE, WAIT UNTIL IT IS           :
                        ;                 DRIVE THERE?                                                :
                        ;                 F  DROP                                                     :
                        ;                    ALL DRIVES DROPPED?                                      :
                        ;                       T DROP MODULE                                         :
                        ;                       F ---                                                 :
                        ;                 T ONLINE                                                    :
                        ;                 ONLINE?                                                     :
                        ;                       T PICK BLOCK - IF RANDOM, GET RAND # MOD X           :
                        ;                                ELSE INCREMENT                               :
                        ;                                       IF LBN > LIMIT THEN LBN <  0          :
                        ;                         WRITE                                               :
                        ;                         READ                                                :
                        ;                         CHECK DATA ?                                        :
                        ;                               T CHECK                                       :
                        ;                         AVAILABLE DRIVE(I)                                  :
                        ;                       F TRY TO BRING ONLINE AGAIN                           :
                        ;               NEXT J                                                        :
                        ;                                                                             :
                        ;******************************************************************************


                        ;******************************************************************************
                        ;                                                                             :
                        ;       START CODE                                                            :
                        ;                                                                             :
                        ;       IF THE CODE IS RESTARTED, CLEAR THE OLD ADDRESSES SO THE              :
                        ;       THE CONTROLLER WILL GET REINITED.                                     :
                        ;                                                                             :
                        ;******************************************************************************

000660                  START:
000660  005227  177777         INC     #-1                     ;FIRST TIME THRU HERE?
000664  001006                 BNE     1$                      ;BR IF NO
000666  042767  000002  177122 BIC     #SR.XFR,SR1             ;DO NOT ALLOW DISK TRANSFERS
000674  104403  000000' 005352' MSGN$,BEGIN,WARN1       ;ASCII MESSAGE CALL WITH COMMON HEADER
000702  032767  000002  177106 1$:    BIT     #SR.XFR,SR1             ;WILL CUSTOMER DATA BE OVERWRITTEN?
000710  001404                 BEQ     2$                      ;BR IF NO
000712  104403  000000' 005356' MSGN$,BEGIN,WARN2       ;ASCII MESSAGE CALL WITH COMMON HEADER
000720  000403                 BR      3$                      ;
000722                  2$:
000722  104403  000000' 005362' MSGN$,BEGIN,WARN3       ;ASCII MESSAGE CALL WITH COMMON HEADER
000730                  3$:
```

MODULE CODE

```
                                        CLR     CDERCT                  ;CLEAR DATA CHECK ERROR COUNT
000730  005067  177210
                                        MOV     #177777,EXPAV           ;NOT EXPECTING AN INTERRUPT
000734  012767  177777  177522
                                        MOV     #PRTNUM,PRNMSG          ;INITIALIZE PRINT WORD
000742  012767  000017  177512
                                        MOV     DVID1,DVICE             ;DVICE HAS DESIRED BITS SET
000750  016767  177040  177622
                                        CLR     TABLEW                  ;SET TABLE FOR UNIT 0
000756  005067  177632
                                        MOV     #1,TABLEW+2             ;SET TABLE FOR PORTID FOR UNIT 0
000762  012767  000001  177626
                                        CLR     CMDREF                  ;COMMAND REF # = 0
000770  005067  177274
                                        RAND$,BEGIN
000774  104417  000000'
                                        MOV     RANNUM,SECL
001000  016767  177050  177560
                                        CLR     SECH                    ; FOR RESTARTING (INITIAL SECTOR ADDR)
001006  005067  177556
                                        MOV     ADDR,SAREG              ; STORE IN SA REG
001012  016767  176770  177232
                                        ADD     #2,SAREG                ; SA REGISTER HAS PROPER ADDRESS
001020  062767  000002  177224
                                        CLR     OLDPA                   ; OLD PHYSICAL ADDRESS CLEARED
001026  005067  177424
                                        CLR     OLDEA                   ; OLD EXTENDED ADDRESS CLEARED
001032  005067  177422
                                                                        ; FOR RESTARTING. THIS WILL FORCE A
                                                                        ; CONTROLLER REINIT TO TAKE PLACE
```

```
                                ;*****************************************************************
                                ;                                                               *
                                ;       RESTART SEQUENCE                                        *
                                ;                                                               *
                                ;       CHECK THE ADDRESS OF THE RINGS TO SEE IF THEY WERE RELOCATED *
                                ;       IF THEY WERE, REINIT THE CONTROLLER.                    *
                                ;                                                               *
                                ;       GET THE NEW ADDRESSES.  IF THE DISKLESS OPERATION IS DESIRED *
                                ;       THEN DO THE MAITENENCE WRITE AND READ.  ELSE DO THE WRITE *
                                ;       AND READ WITH A DRIVE.                                  *
                                ;                                                               *
                                ;*****************************************************************
001036                          RESTRT:
001036  012767  000260' 177376          MOV     @RSPONC,VA              ;DID THE RINGS RELOCATE?
001044  104415  000000' 000442'          GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16 BIT VA
001052  026767  177366  177376          CMP     PA,OLDPA                ;IS THE OLD PHYS ADDR = NEW ONE?
001060  001004                          BNE     RESTR2                  ;IF SO, REINIT
001062  026767  177360  177370          CMP     EA,OLDEA                ;IS THE OLD EXTN ADDR = NEW ONE?
001070  001412                          BEQ     RESTR1                  ;IF NOT, DON'T REINIT
001072  016767  177346  177356  RESTR2: MOV     PA,OLDPA                ;ELSE SET THE OLD RING ADDR
001100  016767  177342  177352          MOV     EA,OLDEA                ; AND THE OLD EXTENED ADDR
001106  004767  000332                  JSR     PC,INITUD               ;AND INIT THE CONTROLLER
001112  005067  177466                  CLR     TRY                     ;CLEAR RETRY COUNT
001116                          RESTR1:
001116  032767  000010  176672          BIT     @SR.SUM,SR1             ;DO WE WANT THE REPORT?
001124  001034                          BNE     1$                      ;IF NOT, SKIP THE REPORT
001126  026767  177330  176700          CMP     PRNMSG,PASCNT           ;DO WE PRINT?
001134  001030                          BNE     1$                      ;IF PASS COUNT IS NOT = PRINT WORD, SKIP
001136  062767  000017  177316          ADD     @PRTNUM,PRNMSG          ;PRINT WORD IS INCREMENT
                                        ;*******************************************************
                                                                ;CONVERT SOFCNT TO ASCII AND
                                                                ;STORE AT ADR2
001144  104421  000000' 000042'         BTOD$,BEGIN,SOFCNT,ADR2
001152  000475'
                                        ;*******************************************************
001154  105067  177322                  CLRB    ADR2+5
                                        ;*******************************************************
                                                                ;CONVERT HRDCNT TO ASCII AND
                                                                ;STORE AT ADR3
001160  104421  000000' 000044'         BTOD$,BEGIN,HRDCNT,ADR3
001166  000504'
                                        ;*******************************************************
001170  105067  177315                  CLRB    ADR3+5
                                        ;*******************************************************
                                                                ;CONVERT CDERCT TO ASCII AND
                                                                ;STORE AT ADR1
001174  104421  000000' 000144'         BTOD$,BEGIN,CDERCT,ADR1
001202  000466'
                                        ;*******************************************************
001204  105067  177263                  CLRB    ADR1+5
001210  104403  000000' 005314'         MSGN$,BEGIN,ERRPAS      ;ASCII MESSAGE CALL WITH COMMON HEADER
001216  012777  004334' 176564  1$:     MOV     @NTRUPT,@VECTOR         ;GET VECTOR ADDRESS
                                                                        ;SET POINTER
001224  104415  000000' 000124'         GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
001232  016700  176672                  MOV     RBUFEA,R0               ;GET EA TO ADJUST
001236  004767  000572                  JSR     PC,ASR04                ;GO ADJUST IT
001242  010067  177202                  MOV     R0,RBFFEA               ;PUT ADJUSTED VALUE IN A SAVE AREA
```

```
MODULE CODE

     43 001246  005067  177330                CLR     UNITNO                  ;PRESET UNIT 0
     44 001252  032767  000002  176536        BIT     @SR.XFR,SR1             ;DISK XFER???
     45 001260  001454                        BEQ     MA10NC                  ;NO!  DO MAINTENENCE (DISKLESS) ROUTINES
                                        ;*****************************************************************
     46                                 ;       DO THE DISK OPERATIONS                                  ;
     47                                 ;               CHECK TO SEE WHICH PORTS ARE AVAILABLE          ;
     48                                 ;*****************************************************************
     49
     50
     51 001262  004767  001032                JSR     PC,SETUP                ;FIND DRIVES/SET UP TABLE
     52 001266  005767  177306                TST     DVICE                   ;ELSE, TEST FOR ANY MORE DRIVES
     53 001272  001002                        BNE     LOOP1                   ;IF TRUE, DO A CYCLE
     54
     55 001274  104410  000000'               END$,BEGIN              ;
     56
     57 001300                          LOOP1:
        001300  104414  000000'               GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
     58 001304  016700  176626                MOV     WBUFEA,R0               ;GET WRITE BUFFER INFORMATION
     59 001310  004767  000520                JSR     PC,ASR04                ;ADJUST IT
     60 001314  010067  177132                MOV     R0,WBFFEA               ;STORE EA IN SAVE AREA
     61 001320  012704  000614'               MOV     @TABLEW,R4              ;R4 -> TABLE OF UNITNO AND PORTID
     62 001324  012703  000001                MOV     @1,R3                   ;R3 IS AN INDEX TO DVICE
     63 001330                          LOOP2:
     64 001330  030367  177244                BIT     R3,DVICE                ;HAS THE DRIVE BEEN DROPPED
     65 001334  001412                        BEQ     9$                      ;IF SO, SKIP THIS DRIVE
     66 001336  016467  000002  177242        MOV     2(R4),PORTID            ;SET UP PORTID
     67 001344  011467  177232                MOV     (R4),UNITNO             ;SET UP UNITNO
     68                                 ; *** DO A DISK CYCLE
     69 001350  004767  001250                JSR     PC,CYCLED               ;DO A CYCLE FOR DISK OPERATION
     70 001354  103002                        BCC     9$                      ;IF SUCCESSFUL, CONTINUE
     71 001356  004767  002026                JSR     PC,DROP1                ;IF NOT, DROP DRIVE
     72 001362                          9$:
     73 001362  062704  000004                ADD     @4,R4                   ;POINT TO NEXT ENTRY OF THE TABLE
     74 001366  006303                        ASL     R3                      ;R3 POINTS TO NEXT BIT
     75 001370  022704  000654'               CMP     @TEND,R4                ; POINT BEYOND LAST ENTRY?
     76 001374  001403                        BEQ     12$                     ; IF NOT, THEN TRY AGAIN.
     77 001376  020367  177176                CMP     R3,DVICE                ;IF R3 > DVICE THEN DONE WITH ITERATION
     78 001402  003752                        BLE     LOOP2                   ;IF < OR =, LOOP
     79 001404                          12$:
     80 001404  104413  000000'               ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                        ;MONITOR SHALL TEST END OF PASS
     81 001410  000733                        BR      LOOP1                   ;AND DO AGAIN
     82
     83
     84                                 ;*****************************************************************
     85                                 ;       MAINTENENCE ROUTINE, DO THE DISKLESS CODE               ;
     86                                 ;*****************************************************************
     87
     88 001412                          MA10NC:
        001412  104414  000000'               GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
     89 001416  016700  176514                MOV     WBUFEA,R0               ;GET EA TO ADJUST
     90 001422  004767  000406                JSR     PC,ASR04                ;ADJUST IT
     91 001426  010067  177020                MOV     R0,WBFFEA               ;STORE EA IN SAVE AREA
     92 001432  004767  001464                JSR     PC,CYCLEL
     93 001436  104413  000000'               ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                        ;MONITOR SHALL TEST END OF PASS
     94 001442  000763                        BR      MA10NC
```

```
                        ;*****************************************************************************
                        ;                                                                            ;
                        ;       INITIALIZE THE CONTROLLER                                            ;
                        ;                                                                            ;
                        ;       DO THE 4 STEPS FOR INITIALIZING THE CONTROLLER.                      ;
                        ;                                                                            ;
                        ;       STEP 1 - CHECK FOR ERROR, STEP 1                                     ;
                        ;                SEND VECTOR/4, INTERRUPT ENABLE, RING LEN'S = 0             ;
                        ;                                                                            ;
                        ;       STEP 2 - CHECK VECTOR ECHO, INTERRUPT ECHO,                          ;
                        ;                        ERROR, STEP 2                                       ;
                        ;                SEND PHYSICAL ADDRESS & PURGE = 0                           ;
                        ;                                                                            ;
                        ;       STEP 3 - CHECK RING LEN = 0, ERROR, STEP 3                           ;
                        ;                SEND EXTENDED ADDRESS BITS                                  ;
                        ;                                                                            ;
                        ;       STEP 4 - CHECK STEP 4                                                ;
                        ;                SEND LFAIL = 0 , GO AND BURST                               ;
                        ;                                                                            ;
                        ;*****************************************************************************
                        
001444  012767  000260' 176770  INITUD: MOV     #RSPONC,VA              ;VA -> RSPONC
001452  104415  000000' 000442'         GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
001460  005004                         CLR     R4                      ;R4 IS USED IF AN ERROR IS DETECTED
001462  012702  000001                 MOV     #1,R2                   ;R2 = STEP INDICATOR REG FOR MSG'S
001466  005077  176314                 CLR     @ADDR                   ;WRITE TO IP REGISTER TO INIT CONTROLLER
001472  012701  002260                 MOV     #TIMER,R1               ;SET TIME OUT LIMIT
001476  017700  176550         1$:     MOV     @SAREG,R0               ;R0 HAS SA REGISTER DATA
001502  032700  100000                 BIT     #<SA.ERR>,R0            ;CHECK FOR ERROR
001506  001007                         BNE     2$                      ;IF FOUND, GET OUT OF LOOP
001510  104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001514  104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001520  005301                         DEC     R1                      ;TIME OUT?
001522  001365                         BNE     1$                      ;IF NOT, LOOP
001524  000404                         BR      4$                      ;IF DONE, CONTINUE
001526  012703  004000         2$:     MOV     #SA.S1,R3               ;R3 = STEP 1 BIT
001532  000167  000412                 JMP     ERROR1                  ;IF HERE, ERROR
001536  042700  001140         4$:     BIC     #<SA.Q22+SA.MAP+SA.SM>,R0; CLEAR KDA50-Q DEPENDENT BITS
001542  022700  004400                 CMP     #<SA.S1+SA.DIA>,R0      ;DID DATA COMPARE PROPERLY?
001546  001402                         BEQ     5$                      ;IF SO, CONTINUE
001550  000167  000370                 JMP     ERROR3                  ;REPORT ERROR
                                ;
                                    STEP 2
001554  016705  176230         5$:     MOV     VECTOR,R5               ;VECTOR GIVEN
001560  006205                         ASR     R5                      ;SET TO APPROPRIATE VALUE
001562  006205                         ASR     R5                      ; = VECTOR/4
001564  052705  100200                 BIS     #<SA.INT+BIT15>,R5      ;ACTIVATE INTERRUPTS & SET MSB FOR STEP 1
                                                                        ;LEN'S ARE 0
001570  010500                         MOV     R5,R0                   ;STORE R5 IN R0 FOR SUBROUTINE
001572  012703  004000                 MOV     #SA.S1,R3               ;R3 HAS STEP BIT FOR SUBROUTINE
001576  004767  000244                 JSR     PC,SNDSTP               ;SEND STEP DATA
001602  042705  100000                 BIC     #BIT15,R5               ;CLEAR MSB FOR COMPARE DATA
001606  042700  000200                 BIC     #BIT07,R0               ;WAS BIT07 ONLY BIT SET?, SHOULD BE
001612  001404                         BEQ     6$
001614  052700  010200                 BIS     #<SA.S2+BIT07>,R0       ;SET R0 TO REPORT THE ERROR
001620  000167  000320                 JMP     ERROR3                  ;REPORT ERROR
001624  016700  176614         6$:     MOV     PA,R0                   ;R0 GETS PHYSICAL ADDRESS
001630  004767  000212                 JSR     PC,SNDSTP               ;SEND STEP DATA
```

```
MODULE CODE

     57 001634  042705  177400                BIC     #177400,R5              ;HIGH BYTE CLEARED
     58 001640  020500                      CMP     R5,R0                   ;CHECK ECHO DATA
     59 001642  001402                      BEQ     7$                      ;IF OK, SKIP
     60 001644  000167  000274              JMP     ERROR3                  ;IF NOT, REPORT ERROR
     61 001650                          7$:
     62                                 ;       STEP 3
     63 001650  016700  176572              MOV     EA,R0                   ;ADJUST THE EXTENDED ADDRESS BITS
     64 001654  004767  000154              JSR     PC,ASR04                ;SHIFT EXTENDED ADDRESS BITS FOR CONTROLLER
     65 001660  004767  000162              JSR     PC,SNDSTP               ;SEND STEP DATA
     66 001664  012700  000254'             MOV     #RSPONC-4,R0            ;R0 -> RING ENVELOP
     67                                 ;       STEP 4
     68 001670  005720                  8$: TST     (R0)+                   ;IS THE RING ENTRY = 0?
     69 001672  001117                      BNE     ERROR5                  ;IF NOT, ERROR
     70 001674  022700  000270'             CMP     #CMDREF,R0              ;IS R0 POINT PAST THE RINGS?
     71 001700  001373                      BNE     8$                      ;IF NOT, LOOP
     72 001702  016700  176112              MOV     SR2,R0                  ;R0 = BURST VALUE
     73 001706  000241                      CLC                             ;CLEAR CARRY
     74 001710  006300                      ASL     R0                      ;ALIGN BURST FOR STEP 4
     75 001712  006300                      ASL     R0                      ; "
     76 001714  052700  000001              BIS     #SA.GO,R0               ;SET GO BET
     77 001720  010077  176326              MOV     R0,@SAREG               ;SEND DATA TO CONTROLLER/INIT DONE
     78 001724  012767  000362' 176510      MOV     #CMPACK,VA              ;GET COMMAND PACKET PA AND EA
     79 001732  104415  000000' 000442'    GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
     80 001740  016767  176500  176316      MOV     PA,COMMND               ;STORE ADDRESS IN THE RING
     81 001746  016700  176474              MOV     EA,R0                   ;SAVE IN R0
     82 001752  004767  000056              JSR     PC,ASR04                ;SHIFT EXTENDED ADDRESS BITS FOR CONTROLLER
     83 001756  010067  176304              MOV     R0,COMMND+2             ;MOVE ADJUSTED EA INTO RING
     84
     85 001762  012767  000276' 176452      MOV     #RSPACK,VA              ;GET RESPONCE PACKET PA AND EA
     86 001770  104415  000000' 000442'    GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
     87 001776  016767  176442  176254      MOV     PA,RSPONC               ;STORE ADDRESS IN THE RING
     88 002004  016700  176436              MOV     EA,R0                   ;SAVE IN R0
     89 002010  004767  000020              JSR     PC,ASR04                ;SHIFT EXTENDED ADDRESS BITS FOR CONTROLLER
     90 002014  010067  176242              MOV     R0,RSPONC+2             ;MOVE ADJUSTED EA INTO RING
     91 002020  012777  004334' 175762      MOV     #NTRUPT,@VECTOR         ;STORE INTERRUPT ADDRESS IN VECTOR
     92 002026  005067  176552              CLR     TRY                     ;CLEAR TRY SO DRIVE WILL
     93                                                                     ;GO BACK ONLINE IF NECESSARY
     94 002032  000207                      RTS     PC
     95
     96                                 ;***************************************************************************
     97                                 ;                                                                          *
     98                                 ;       ASR04                                                              *
     99                                 ;               ARITHMETIC SHIFT RIGHT REG 0 FOUR TIMES                    *
    100                                 ;                                                                          *
    101                                 ;       EXTENDED ADDRESS BITS (16 & 17) ARE SET IN BIT POSITION 4 & 5      *
    102                                 ;       RESPECTIVELY.  SHIFT RIGHT FOUR TIMES TO REPOSTION THE VALUE       *
    103                                 ;                                                                          *
    104                                 ;       INPUT   R0 = UNADJUSTED EXTENDED ADDRESS BITS                      *
    105                                 ;                                                                          *
    106                                 ;       OUTPUT  R0 = ADJUSTED EXTENDED ADDRESS BITS                        *
    107                                 ;                                                                          *
    108                                 ;***************************************************************************
    109 002034                          ASR04:
    110 002034  006200                      ASR     R0                      ;SHIFT 10
    111 002036  006200                      ASR     R0                      ; SHIFT 4
    112 002040  006200                      ASR     R0                      ;  SHIFT 2
    113 002042  006200                      ASR     R0                      ;   SHIFT 1
```

```
002044  000207                           RTS     PC                  ;RETURN

                                 ;*****************************************************************
                                 ;
                                 ;       SEND STEP DATA
                                 ;
                                 ;       INPUT:  R0 HAS DATA TO BE SENT TO CONTROLLER FOR STEP
                                 ;               R3 HAS PREVIOUS STEP FLAG SET
                                 ;
                                 ;       OUTPUT: R0 HAS DATA SENT FROM CONTROLLER TO HOST FOR ECHO AND NEXT STEP
                                 ;               R3 HAS CURRENT STEP FLAG SET
                                 ;
                                 ;*****************************************************************
002046  016701  175736           SNDSTP: MOV     VECTOR,R1               ;
002052  012721  002072'                  MOV     #INTA,(R1)+             ;SET UP INTERRUPT HANDLER ADDRESS
002056  116711  175730                   MOVB    BR1,(R1)                ;SET PRIORTITY LEVEL
002062  010077  176164                   MOV     R0,@SAREG               ;SEND STEP1 WRITE FORMMATED DATA

002066  104400  000000'                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

002072                           INTA:
                                         ;------------- --------- -------- ------ -----------------   -
002072  000004  000000' 002100'          PIRQ$,BEGIN,3$          ; QUEUE UP TO CONTINUE AT 3$ AND RTI
                                         ;------ ------------ - ------------------ ---- ----- - --
002100                           3$:
002100  017700  176146                   MOV     @SAREG,R0               ;GET STEP N FORMATTED DATA
002104  032700  100000                   BIT     #SA.ERR,R0              ;TEST FOR ERROR
002110  001017                           BNE     ERROR1                  ;IF NOT OK, REPORT
002112  005202                           INC     R2                      ;SET STEP REGISTER
002114  006303                           ASL     R3                      ;R3 HAS STEP BIT PROPERLY SET
002116  030300                           BIT     R3,R0                   ;WAS STEP N SET?
002120  001002                           BNE     4$                      ;IF SO, CONTINUE
002122  000167  000020                   JMP     ERROR2                  ;IF NOT CORRECT STEP, ERROR
002126  040300                   4$:     BIC     R3,R0                   ;CLEAR THE STEP BIT, FOR COMPARE
002130  000207                           RTS     PC                      ;RETURN
```

```
                        ;*****************************************************************************
                        ;
                        ;       ERROR 1
                        ;       PRINT AN ERROR REPORTED BY THE CONTROLLER DIAGNOSTICS
                        ;
                        ;       ERROR2
                        ;       PRINT THE VALUE OF THE SA REGISTER WHEN THE STEP BIT WAS NOT SET
                        ;
                        ;       ERROR3
                        ;       PRINT A THE VALUE OF THE SA REGISTER WHEN THE ECHO WAS NOT SET
                        ;               CORRECTLY
                        ;
                        ;       INPUT   R0 -> SA REGISTER
                        ;               R2 = STEP COUNT
                        ;
                        ;       OUTPUT  THE RETRY COUNT IS INCREMENTED
                        ;               IF THE RETRY COUNT > RETRY LIMIT, END MODULE
                        ;
                        ;       ERROR5
                        ;       RING WASN'T ALL ZERO -> ERROR
                        ;       DROP UDBA0
                        ;
                        ;*****************************************************************************
                        ERROR5:
002132                          MSGN$,BEGIN,ZERO        ;ASCII MESSAGE CALL WITH COMMON HEADER
002132  104403  000000' 005372'
002140  104410  000000'          END$,BEGIN              ;
002144  005204                  ERROR3: INC     R4              ;R4 = 3 FOR ERROR3
002146  005204                  ERROR2: INC     R4              ;R4 = 2 FOR ERROR2
002150  005204                  ERROR1: INC     R4              ;R4 = 1 FOR ERROR1
002152  010267  176276                  MOV     R2,NUM          ;STORE STEP REG IN A NUMBER FOR CONVRT
                                        ;*************************************************************
                                                                ;CONVERT NUM TO ASCII AND
                                                                ;STORE AT ADR2
002156  104420  000000' 000454'         OTOA$,BEGIN,NUM,ADR2
002164  000475'
                                        ;*************************************************************
002166  017767  176060  176260          MOV     @SAREG,NUM      ;STORE VALUE IN A NUMBER
                                        ;*************************************************************
                                                                ;CONVERT NUM TO ASCII AND
                                                                ;STORE AT ADR1
002174  104420  000000' 000454'         OTOA$,BEGIN,NUM,ADR1
002202  000466'
                                        ;*************************************************************
002204  005304                          DEC     R4              ;ERROR 1?
002206  001003                          BNE     1$              ;IF NOT, CHECK IF IT IS THE NEXT ERROR
002210  104403  000000' 005202'         MSGN$,BEGIN,INITE1      ;ASCII MESSAGE CALL WITH COMMON HEADER
002216                          1$:
002216  005304                          DEC     R4              ;ERROR 2?
002220  001003                          BNE     2$              ;IF NOT, CHECK IF IT IS THE NEXT ERROR
002222  104403  000000' 005226'         MSGN$,BEGIN,INITE2      ;ASCII MESSAGE CALL WITH COMMON HEADER
002230                          2$:
002230  005304                          DEC     R4              ;ERROR 3?
002232  001003                          BNE     3$              ;IF NOT, CHECK IF IT IS THE NEXT ERROR
002234  104403  000000' 005234'         MSGN$,BEGIN,INITE3      ;ASCII MESSAGE CALL WITH COMMON HEADER
```

MODULE CODE

```
                                3$:
                                        ;*****************************************************************
                                                                ;CONVERT ADDR TO ASCII AND
                                                                ;STORE AT ADR3
                                        OTOA$,BEGIN,ADDR,ADR3

                                        ;*****************************************************************
                                        ;*****************************************************************
                                        HRDER$,BEGIN,NULL       ;
                                        ;*****************************************************************
                                        MSGN$,BEGIN,INITER      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        INC     TRY                     ;INCREMENT RETRY COUNT
                                        CMP     #RLIM,TRY               ;IS THE RETRY COUNT EXCEEDED?
                                        BEQ     6$                      ;IF SO, END IT
                                        JMP     START                   ;IF NOT, TRY AGAIN
                                6$:
                                        MSGN$,BEGIN,ABORT       ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        END$,BEGIN              ;
```

```
                                ;*****************************************************************************
                                ;
                                ;       SET UP
                                ;
                                ;       GO FIND OUT WHAT DRIVES ARE OUT THERE.
                                ;       A TABLE IS FILLED WITH UNIT NUMBERS(MAX IS 16)
                                ;
                                ;       THIS SHOULD ONLY BE DONE AT THE VERY BEGINNING OF RUNNING
                                ;       THIS DECX MODULE; THEN NOT RUN AGAIN.
                                ;
                                ;       INPUT:  DVICE HAS APPROPRIATE BITS SET.  THE # OF BITS =
                                ;                       # OF DRIVES WANTED TO TEST.
                                ;               POSITION OF BITS = WHICH DRIVE IN THE SYSTEM IS DESIRED.
                                ;
                                ;*****************************************************************************
002320                          SETUP:
                                ; ***   SET CONTRL CHAR AND WAIT FOR THE ATTENTION MESSAGES
002320  004767  001650                  JSR     PC,SCC                  ;SET CONTROLLER CHARACTERISTICS
002324  005367  175740                  DEC     CMDREF                  ;ONLY SET UP AT BEGINNING OF MODULE
002330  001110                          BNE     19$                     ; (USE DRIVES FOUND AT BEGINNING)
002332  012703  000001                  MOV     #1,R3                   ;INITIAL PORTID VALUE
002336  012704  000614'                 MOV     #TABLEW,R4              ;R4 -> TABLEW
002342  011467  176234                  MOV     (R4),UNITNO             ;INITIAL UNITNO IN TABLEW
002346  016714  176230          1$:     MOV     UNITNO,(R4)             ;UNIT NO SET IN TABLEW;READY TO TEST
002352  010367  176230                  MOV     R3,PORTID               ;PORT ID SET
002356  010364  000002                  MOV     R3,2(R4)                ;PORTID SET IN TABLEW
002362  012764  177777  000004          MOV     #177777,4(R4)           ;INSERT NEW -1,-1 FOR LAST ENTRY
002370  016464  000004  000006          MOV     4(R4),6(R4)             ;  OF THE TABLEW
002376  012767  002400  176206          MOV     #2400,WORK              ;WORK = RETRY LIMIT
002404  004767  001526          3$:     JSR     PC,GTSTAT               ;GET STATUS, GET NEXT UNIT NUMBER
002410  103006                          BCC     7$                      ;OK, CONTINUE
002412  005367  176174                  DEC     WORK                    ;ELSE IF OFFLINE, DECR COUNT
002416  001372                          BNE     3$                      ;IF COUNT > 0, TRY AGAIN.
002420  004767  000774          5$:     JSR     PC,DROP2                ;DROP THE DRIVE
002424  000437                          BR      17$                     ;TRY NEXT UNIT
002426  016767  175650  176146  7$:     MOV     P.UNIT+RSPACK,UNITNO    ;UNIT NUMBER FROM RESPONCE PACKET IN UNITNO
                                ; *** CHECK FOR CASE WHERE THE MORE UNITS THEN DRIVES HAVE BEEN SPECIFIED.
                                ; *** NEXT UNIT MODIFIER WILL GIVE A DUPLICATE UNIT NUMBER.
002434  012702  000614'                  MOV     #TABLEW,R2              ;R2 -> TABLE TO FIND DUPLICATE
002440  012705  000001                  MOV     #1,R5                   ;R5 IS TEMP PORTID
002444  020227  000654'          9$:     CMP     R2,#TEND                ;REACHED THE BOTTOM?
002450  001420                          BEQ     15$                     ;IF SO, EXIT
002452  020305                          CMP     R3,R5                   ;REACHED THE LATEST ENTRY?
002454  001416                          BEQ     15$                     ;IF SO, EXIT
002456  026712  176120                  CMP     UNITNO,(R2)             ;DO WE HAVE A DUPLICATE UNIT NUMBER?
002462  001404                          BEQ     13$                     ;IF SO, ERROR
002464  062702  000004          11$:    ADD     #4,R2                   ;IF NOT, POINT TO NEXT POINTER
002470  006305                          ASL     R5
002472  000764                          BR      9$                      ;AND CONTINUE
002474  011467  176102          13$:    MOV     (R4),UNITNO             ;DROP DRIVE FROM TABLE
002500  010367  176102                  MOV     R3,PORTID               ;
002504  004767  000720                  JSR     PC,DROP3                ;AND DROP IT
002510  000405                          BR      17$
002512                          15$:
                                ; ***
002512  026714  176064                  CMP     UNITNO,(R4)             ;IS THE UNITNO CORRECT?
002516  001402                          BEQ     17$                     ;IF SO, CHECK FOR NEXT UNIT
```

```
MODULE CODE

     58 002520  016714  176056                        MOV     UNITNO,(R4)             ;ELSE, CORRECT THE UNIT NUMBER IN TABLE
     59 002524                              17$:
     60 002524  006303                              ASL     R3                      ;NEXT PORTID SET
     61 002526  026703  176046                      CMP     DVICE,R3                ;DONE?
     62 002532  100407                              BMI     19$                     ;IF R3 > DVICE, ALL DESIRED DRIVES ARE FOUND
     63 002534  005267  176042                      INC     UNITNO                  ;NEXT UNITNO SET
     64 002540  062704  000004                      ADD     #4,R4                   ;POINT TO NEXT ENTRY TO TEST DRIVE
     65 002544  022704  000654'                     CMP     #TEND,R4                ;POINT TO END? IF SO, TABLE FULL
     66 002550  101276                              BHI     1$                      ;IF R4 NOT REACHED END, GO TEST
     67 002552                              19$:
     68 002552  000207                              RTS     PC
     69
     70                                     ;*****************************************************************************
     71                                     ;
     72                                     ;       TSTOFL
     73                                     ;               TEST TO SEE WHAT KIND OF AN OFFLINE CONDITION HAS OCCURED.
     74                                     ;
     75                                     ;*****************************************************************************
     76 002554  022700  000003              TSTOFL: CMP     #ST.OFL,R0              ;WAS THE DRIVE FOUND OFFLINE?
     77 002560  001403                              BEQ     10$                     ;CHECK WHAT KIND OF OFFLINE
     78 002562  022700  000013                      CMP     #ST.DRV,R0              ;WAS IT A DRIVE ERROR? -> SDI?
     79 002566  001012                              BNE     13$                     ;IF IT WAS NOT, ERROR (DROP DRIVE)
     80 002570  032767  000740  175512      10$:    BIT     #<SC.NVL+SC.DIS+SC.DUP+SC.IOP>,P.STS+RSPACK     ;WERE ANY OF THESE BITS SET?
     81                                                                             ; = NO VOLUME MOUNTED, UNIT DIABLED BY FIELD SREVICE
     82                                                                             ;   OR DUPLICATE UNIT NUMBER OR UNIT INOPERATIVE
     83 002576  001004                              BNE     12$                     ;IF SO, EXIT
     84 002600  032767  177000  175502              BIT     #^C<SC.NVL+SC.DIS+SC.DUP+SC.IOP+ST.MSK>,P.STS+RSPACK ; ANY OTHER DATA?
     85 002606  001002                              BNE     13$                     ;IF SO, DROP
     86 002610  000241                      12$:    CLC                             ;CLEAR CARRY
     87 002612  000207                              RTS     PC                      ;RETURN
     88 002614  000261                      13$:    SEC                             ;SET CARRY, DRIVE WAS FOUND TO BE OFFLINE
     89                                                                             ; OR ANOTHER ERROR
     90 002616  004767  002124                      JSR     PC,ERRORH               ;REPORT ERROR
     91 002622  000207                              RTS     PC                      ;RETURN
```

```
                        ;*****************************************************************************
                        ;                                                                            :
                        ;       CYCLE DISK                                                           :
                        ;                                                                            .
                        ;       DO THE DISK CYCLE                                                    .
                        ;               DO GET STATUS COMMANDS TO ASSURE THAT THE DRIVE              .
                        ;                       IS AVAILABLE (FOR DUAL PORTING)                      ;
                        ;               CHECK DRIVE TO BE ONLINE                                     :
                        ;               IF TRUE                                                      .
                        ;                    PICK THE BLOCK                                          :
                        ;                    WRITE                                                   :
                        ;                    READ                                                    :
                        ;                    DATA CHECK                                              :
                        ;                    MAKE THE DRIVE AVAILABLE                                :
                        ;               ELSE DROP DRIVE                                              :
                        ;                                                                            :
                        ;*****************************************************************************
002624                  CYCLED:
002624  032767  001000  175164          BIT     @SR.DUA,SR              ;DUAL PORT?
002632  001004                          BNE     2$                      ;IF NOT, CONTINUE
                        ; *** CHECK IF WE DO ONLINE FOR THE FIRST TIME.
002634  005767  175744                  TST     TRY                     ;IF TRY HAS SET MSB, DON'T DO ONLINE
002640  100443                          BMI     16$                     ; DON'T DO ONLINE
002642  000422                          BR      10$                     ; ELSE DO ONLINE (1ST TIME THROUGH IN THIS PASS)
                        ; ***
                        ; *** DO GET STATUS COMMANDS TO ASSURE THE DRIVE IS AVAILABLE TO THE CONTROLLER
                        ; ***  FOR DUAL PORTING.
                        ; ***
002644  012701  000010  2$:     MOV     @10,R1                  ;R1 = @ OF GET STATUS TO DO
002650  004767  001262  4$:     JSR     PC,GTSTAT               ;IS THE DRIVE OFFLINE?
002654  103013                          BCC     6$              ;IF ALL OK, DO THE CYCLE
002656  004767  177672                  JSR     PC,TSTOFL       ;ELSE, CHECK IF OFFLINE
002662  103507                          BCS     24$             ;IF IT ERRED, DROP THE DRIVE
                        ; *** HANDLE OFF LINE DRIVE, WAIT FOR AVAILABLE ATTENTION MESSAGE
002664  005067  175574                  CLR     EXPAV           ;EXPECT AN AVAILABLE ATTENTION MESSAGE
002670  052767  140000  175364          BIS     @<RG.OWN+RG.FLG>,RSPONC+2       ;SET RING FOR ATTN MESSAGE
002676  004767  001426                  JSR     PC,INTERP       ;WAIT FOR MESSAGE
                                                                ; 2ND ATTENTION MESSAGE
002702  000402                          BR      10$
002704  005301                  6$:     DEC     R1              ;DONE?
002706  001360                          BNE     4$              ;IF NOT DONE, TRY AGAIN
002710  004767  001316          10$:    JSR     PC,ONLINE       ;DO AND ONLINE COMMAND
002714  103753                          BCS     2$              ;IF CARRY WAS SET, TRY AGAIN
002716  016767  175422  175650  14$:    MOV     P.UNSZ+2+RSPACK,UNSZH   ;IS THE UNIT SIZE HI ADDRESS
002724  016767  175412  175640          MOV     P.UNSZ+RSPACK,UNSZL     ;GET UNIT SIZE/IS IT = 0?
002732  001006                          BNE     16$             ;IF NOT ZERO, CONTINUE WITH ITERATION
002734  005767  175634                  TST     UNSZH           ;IS UNSZH ALSO 0?
002740  001731                          BEQ     CYCLED          ;IF 0, TRY TO BRING ONLINE AGAIN
                        ; *** SET MSB OF TRY TO SHOW THAT INITIAL ONLINE IS DONE
002742  012767  100000  175634          MOV     @100000,TRY

                        ;*****************************************************************************
                        ;                                                                            :
                        ;       THE FOLLOWING SEQMENT SETS THE LIMIT FOR THE UNIT SIZE.              :
                        ;       THE VALUE (UNIT SIZE - (WRITE BUFFER SIZE/NORMAL BLOCK SIZE))        :
                        ;       IS THE LAST SECTOR POSSIBLE TO RIGHT TO.                             .
                        ;                                                                            .
```

```
                                        ;*****************************************************************
   002750                               16$:
   002750 016700  175166                        MOV     WBUFSZ,R0               ;WBUFSZ IN R0 AS A LIMIT
   002754 005001                                CLR     R1                      ;R1 = # OF BLOCKS
   002756 005201                        18$:    INC     R1                      ;INCREMENT THE # OF BLOCKS
   002760 162700  000400                        SUB     #400,R0                 ;DECREMENT A BLOCK
   002764 100374                                BPL     18$                     ;BR IF > 0
   002766 160167  175600                        SUB     R1,UNSZL                ;ADJUST THE UNIT SIZE
                                        ; ***   NOW PICK WHICH BLOCK TO WRITE TO
   002772 004767  000156                        JSR     PC,PICKBK               ;ELSE SELECT A SECTOR TO TEST
   002776 004767  001134                        JSR     PC,GTSTAT               ;DID WE NOT GET THE DRIVE ONLINE?
   003002 103720                                BCS     2$                      ;IF WE DID NOT, GO BACK TO TOP AND TRY AGAIN
   003004 022700  000004                        CMP     #ST.AVL,R0              ;IS IT AVAILABLE?
   003010 001715                                BEQ     2$                      ;IF SO, GO BACK TO TOP AND TRY AGAIN
                                        ; ***   WRITE TO THE BLOCK SELECTED
   003012 004767  000720                        JSR     PC,WRITE                ;WRITE THE DATA FOR USER DEFINED # OF WORDS
   003016 103007                                BCC     19$                     ;IF OK, CONTINUE
   003020 032767  001000  174770                BIT     #SR.DUA,SR1             ;ARE WE DOING DUAL PORT?
   003026 001306                                BNE     2$                      ;IF YES, RETRY
   003030 004767  001712                        JSR     PC,ERRORH               ;ELSE, HARD ERROR
   003034 000421                                BR      22$                     ;AND EXIT; BCS 22$ ;IF ERROR, EXIT
                                        ; ***   READ IT BACK
   003036 004767  000730                19$:    JSR     PC,READ                 ;READ A BLOCK
   003042 103416                                BCS     22$                     ;IF ERROR, EXIT
   003044 032767  004000  174744                BIT     #SR.CMP,SR1             ;DO A DATA COMPARE?
   003052 001004                                BNE     20$                     ;IF NOT, SKIP THE COMPARE
                                        ; ***   COMPARE DATA
   003054 104412  000000' 000126'               CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
   003062 003064'                               .+2                     ; IF ERROR, CONTINUE
   003064 032767  001000  174724        20$:    BIT     #SR.DUA,SR1             ;DO WE DO AN AVAILABLE?
   003072 001402                                BEQ     22$                     ;IF NOT(BIT NOT SET) SKIP AVAILABLE
                                        ; ***   MAKE THE DRIVE AVAILABLE
   003074 004767  001014                        JSR     PC,AVAILB               ;RELEASE THE DRIVE
   003100 000241                        22$:    CLC                             ;EVERY THING WAS OK
                                                                                ;WASTE A LITTLE TIME SO OTHER
                                                                                ;  CONTROLLER CAN GRAB DRIVE
   003102 000207                        24$:    RTS     PC                      ;RETURN

                                        ; *** SUBROUTINE TO WAIT FOR AN INTERRUPT
                                        ; *** RETURNS AFTER THE INTERRUPT OCCURS
   003104                               DOINTR:
   003104 005067  175354                        CLR     EXPAV                   ;EXPECT AN AVAILABLE ATTENTION MESSAGE
   003110 052767  140000  175144                BIS     #<RG.OWN+RG.FLG>,RSPONC+2   ; SET OWN AND FLAG FOR RESPONCE RING
   003116 000167  001206                        JMP     INTERP          ;WAIT FOR ATTENTICN MESSAGE & RETURN

                                        ;*****************************************************************
                                        ;                                                               ;
                                        ;       DISKLESS CYCLE                                          ;
                                        ;               DO A MAITENENCE WRITE                           ;
                                        ;               AND A MAITENENCE READ                           ;
                                        ;               AND CHECK THE DATA                              ;
                                        ;                                                               ;
                                        ;*****************************************************************
   003122                               CYCLEL:
   003122 004767  000470                        JSR     PC,MAITW                ;DO A MAINTENENCE WRITE
   003126 004767  000430                        JSR     PC,MAITR                ;DO A MAINTENENCE READ
   003132 032767  004000  174656                BIT     #SR.CMP,SR1             ;DO A DATA COMPARE?
```

MODULE CODE

```
    114 003140  001004                          BNE     21$                     ;IF NOT, SKIP THE COMPARE
    115 003142  104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
        003150  003152'                          .+2                     ; IF ERROR, CONTINUE
    116 003152                          21$:
    117 003152  000207                          RTS     PC
```

```
                                ;*****************************************************************************
                                ;                                                                            *
                                ;       PICK A BLOCK TO WRITE TO.                                            *
                                ;                                                                            *
                                ;       EITHER PICK THE NEXT SEQUNTIAL BLOCK (DEFAULT) OR TAKE ONE AT        *
                                ;       RANDOM.                                                              *
                                ;                                                                            *
                                ;       OUTPUT: FILL SECH & SECL (CURRENT SECTOR ADDR)                       *
                                ;                                                                            *
                                ;*****************************************************************************
003154                          P1CKBK:
003154  032767  002000  174634          BIT     #SP.SEQ,SR1             ;CHECK SR1 FOR RANDOM ACCESS MODE
003162  001467                          BEQ     SEQACC                  ;BR IF SEQUENTIAL ACCESS
003164                          RANACC:
003164  104417  000000'                 RAND$,BEGIN
003170  016746  174660                  MOV     RANNUM,-(SP)            ;GENERATE THE SECTOR ADDRESS
003174  104417  000000'                 RAND$,BEGIN
003200  016746  174650                  MOV     RANNUM,-(SP)            ;GENERATE THE SECTOR ADDRESS
                                ;
                                ;       ADJUST HI ADDRESS FIRST
                                ;
003204  000241                          CLC                             ;CLEAR CARRY FOR ROTATE
003206  042716  100000                  BIC     #100000,(SP)            ;CLEAR UPPER BIT MAKES SURE VALUE'S +
003212  012667  175352                  MOV     (SP)+,SECH              ;STORE IN SECTOR HI ADDRESS
003216  005767  175352                  TST     UNSZH                   ;IS THE MAX SIZE 0?
003222  001430                          BEQ     3$                      ;IF 0, GET LOW SECTOR ADDRESS
                                ;  ***  UNSZH > 0 IF CODE FALLS THROUGH HERE
003224  016700  175344                  MOV     UNSZH,R0                ;R0 = MAX VALUE
003230  005100                          COM     R0                      ;R0 COMPLEMENT, NOW FIND MS ZERO
003232  012701  100000                  MOV     #100000,R1              ;R1 IS INDEX INTO MAX VALUE
003236  030100                  1$:     BIT     R1,R0                   ;HAVE 0 YET?
003240  001403                          BEQ     2$                      ;IF 1ST 0 REACHED, CLEAR REST OF THE BITS
003242  000241                          CLC                             ;CLEAR CARRY FOR ROR
003244  006001                          ROR     R1                      ;POINT TO NEXT BIT
003246  000773                          BR      1$                      ;BRANCH TO TEST AGAIN
003250  040100                  2$:     BIC     R1,R0                   ;CLEAR REST OF THE BITS
003252  000241                          CLC                             ;CLEAR CARRY FOR ROR
003254  006001                          ROR     R1                      ;IF R1 ROTATES INTO CARRY, R1 = 0
003256  001374                          BNE     2$                      ;IF R1 NOT 0, MORE BITS TO CLEAR
003260  040067  175304                  BIC     R0,SECH                 ;CLEAR UPPER BITS OF HIGH SECTOR VALUE
003264  026767  175300  175302          CMP     SECH,UNSZH              ;IF THE HIGH SECTOR VALUE > MAX VALUE?
003272  002420                          BLT     7$                      ;IF <, EXIT
003274  001405                          BEQ     4$                      ;IF =, TEST LOW ORDER VALUE
003276  006267  175266                  ASR     SECH                    ;SECH = SECH/2   CAN'T BE > MAX NOW
003302  000414                          BR      7$                      ;EXIT
                                ;
                                ;       GET LOW SECTOR ADDRESS
                                ;
003304  005067  175260          3$:     CLR     SECH                    ;CLEAR HI SECTOR SIZE
003310  005767  175256          4$:     TST     UNSZL                   ;IS THE HIGHEST POSSIBLE = 0?
003314  001406                          BEQ     6$                      ;IF TRUE, DON'T DO LOOP
003316  166716  175250          5$:     SUB     UNSZL,(SP)              ;ELSE, SECL = SECL - UNSZL (ADJUST)
003322  103375                          BCC     5$                      ;IF UNSZL > SECL, LOOP
003324  066716  175242                  ADD     UNSZL,(SP)              ;ELSE SUBTRACTED ONCE TOO OFTEN
003330  000401                          BR      7$                      ; AND EXIT
003332  005016                  6$:     CLR     (SP)                    ;CLEAR LO SECTOR ADDRESS (IF HIGHEST POSSIBLE = 0)
003334  012667  175226          7$:     MOV     (SP)+,SECL              ;SAVE LO SECTOR ADDRESS
```

```
     58 003340  000207                          RTS     PC                      ; RETURN
     59
     60
     61                                 ;GENERATE DISK ADDRESS BY SEQUENTIAL ADDRESSING
     62
     63 003342                          SEQACC:
     64 003342  005267  175220                   INC     SECL                    ;INCREMENT THE SECTOR ADDRESS
     65 003346  001405                           BEQ     16$                     ;BR IF ZERO
     66 003350  026767  175212  175214           CMP     SECL,UNSZL              ;OVER LIMIT?
     67 003356  103413                           BLO     18$                     ;BR IF LOWER
     68 003360  000402                           BR      17$                     ;SKIP THE INCREMENT
     69 003362                          16$:
     70 003362  005267  175202                   INC     SECH                    ;INCREMENT SECTOR HIGH ADDRESS
     71 003366                          17$:
     72 003366  026767  175176  175200           CMP     SECH,UNSZH              ;OVER LIMIT?
     73 003374  103404                           BLO     18$                     ;BR IF LOWER
     74 003376  005067  175164                   CLR     SECL                    ;RESET THE STARTING SECTOR ADDRESS
     75 003402  005067  175162                   CLR     SECH                    ;
     76
     77 003406                          18$:
     78 003406  000207                           RTS     PC
```

```
                        ;*****************************************************************************
                        ;                                                                            *
                        ;       DROP A DRIVE                                                         *
                        ;                                                                            *
                        ;       A DRIVE WOULDN'T RESPOND, DROP IT.  SET THIS UP IN DVICE.            *
                        ;                                                                            *
                        ;       INPUT   UNITNO = UNIT NUMBER OF DRIVE TO DROP                        *
                        ;               PORTID = BIT SET TO DROP DRIVE                               *
                        ;                                                                            *
                        ;       OUTPUT  DVICE HAS A BIT CLEARED.  THE BIT POSITION                   *
                        ;               REPRESENTS THE DRIVE                                         *
                        ;                                                                            *
                        ;*****************************************************************************

003410                  DROP1:
003410  012767  000001  175036          MOV     #1,NUM
003416  000407                          BR      DROP4
003420                  DROP2:
003420  012767  000002  175026          MOV     #2,NUM
003426  000403                          BR      DROP4
003430                  DROP3:
003430  012767  000003  175016          MOV     #3,NUM
003436  036767  175144  175134  DROP4:  BIT     PORTID,DVICE            ;HAS THE DRIVE BEEN DROPPED, DON'T DROP AGAIN
003444  001445                          BEQ     10$                     ;
003446  022767  177777  175132          CMP     #177777,PORTID          ;IF DRIVE HAS BEEN DROPPED, DON'T DROP AGAIN
                                                                        ;       (WILL ZERO DVICE PREMATURE)
003454  001441                          BEQ     10$                     ;IF =, DRIVE HAS BEEN DROPPED  > EXIT ROUTINE

003456  046767  175124  175114          BIC     PORTID,DVICE            ;DROP THE DRIVE
                                        ;***************************************************************
                                                                        ;CONVERT UNITNO TO ASCII AND
                                                                        ;STORE AT ADR2
003464  104421  000000' 000602'         BTOD$,BEGIN,UNITNO,ADR2
003472  000475'
                                        ;***************************************************************
003474  105067  175002                  CLRB    ADR2+5
                                        ;***************************************************************
                                                                        ;CONVERT PORTID TO ASCII AND
                                                                        ;STORE AT ADR1
003500  104420  000000' 000606'         OTOA$,BEGIN,PORTID,ADR1
003506  000466'
                                        ;***************************************************************
003510  012764  177777  000002          MOV     #177777,2(R4)           ;DESELECT DRIVE SO IT WON'T BE USED AGAIN.
003516  005367  174732                  DEC     NUM                     ;DROPPED FOR WHICH ERROR?
003522  001004                          BNE     1$                      ;IF NOT FOR ERRORS, CONTINUE
003524  104403  000000' 005242'         MSGN$,BEGIN,DRP1        ;ASCII MESSAGE CALL WITH COMMON HEADER
003532  000412                          BR      10$
003534  005367  174714          1$:     DEC     NUM                     ;WAS UNIT NOT FOUND?(NON EXISTENT UNIT)
003540  001004                          BNE     2$                      ;IF NOT, CONTINUE
003542  104403  000000' 005260'         MSGN$,BEGIN,DRP2        ;ASCII MESSAGE CALL WITH COMMON HEADER
003550  000403                          BR      10$
003552                          2$:
003552  104403  000000' 005276'         MSGN$,BEGIN,DRP3        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                                                        ; ACTUAL UNITS FOUND
003560  000207                  10$:    RTS     PC
```

```
47
```

```
                                ;*****************************************************************************
                                ;                                                                            ;
                                ;       MAITENENCE READ                                                      ;
                                ;                                                                            ;
                                ;       SET UP A PACKET WITH:                                                ;
                                ;               OPCODE & MODIFIER                                            ;
                                ;               REGION ID & REGION OFFSET                                    ;
                                ;               READ BUFFER DESCRIPTOR                                       ;
                                ;               BYTE COUNT                                                   ;
                                ;       THEN SEND THE PACKET                                                 ;
                                ;                                                                            ;
                                ;*****************************************************************************
003562  004767  001066          MAITR:  JSR     PC,CLRPAK               ;CLEAR THE PACKETS
003566  012767  000030  174576          MOV     #OP.MRD,P.OPCD+CMPACK   ;SET THE OPCODE
003574  016767  174650  174602          MOV     RBFFEA,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR
003602  016767  174320  174572          MOV     RBUFPA,P.ADPA+CMPACK    ;
003610  016700  174316                  MOV     RBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
003614  000424                          BR      MAITP                   ;SET UP THE REST OF THE PACKET

                                ;*****************************************************************************
                                ;                                                                            ;
                                ;       MAITENENCE WRITE                                                     ;
                                ;                                                                            ;
                                ;       SET UP A PACKET WITH:                                                ;
                                ;               OPCODE & MODIFER                                             ;
                                ;               REGION ID & REGION OFFSET                                    ;
                                ;               WRITE BUFFER DESCRIPTOR                                      ;
                                ;               BYTE COUNT (EITHER WBUFSZ OR LIMIT IF WBUFSZ > LIMIT)        ;
                                ;       THEN SEND THE PACKET                                                 ;
                                ;                                                                            ;
                                ;*****************************************************************************

003616  004767  001032          MAITW:  JSR     PC,CLRPAK               ;CLEAR THE PACKETS
003622  012767  000031  174542          MOV     #OP.MWR,P.OPCD+CMPACK   ;SET THE OPCODE
003630  016767  174616  174546          MOV     WBFFEA,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR
003636  016767  174272  174536          MOV     WBUFPA,P.ADPA+CMPACK    ;
003644  026767  174272  174724          CMP     WBUFSZ,LIMIT            ;IS THE BUFFER SIZE > LIMIT?
003652  100403                          BMI     1$                      ;IF NOT, WBUFSZ IS OK
003654  016700  174716                  MOV     LIMIT,R0                ;STORE THE BUFFER SIZE IN WORDS
003660  000402                          BR      MAITP                   ;AND SKIP
003662  016700  174254          1$:     MOV     WBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
003666  006300                  MAITP:  ASL     R0                      ;MAKE IT NUMBER OF BYTES
003670  010067  174502                  MOV     R0,P.BCNT+CMPACK        ;SET WRITE BUFFER SIZE
003674  012767  000020  174370          MOV     #16.,RSPLEN             ;SET RESPONCE PACKET LENGTH
003702  012767  000044  174446          MOV     #36.,CMPLEN             ;SET COMMAND PACKET LENGTH
003710  012767  000001  174500          MOV     #1,P.RGID+CMPACK        ;SET REGION ID = 1
003716  012767  177777  174434          MOV     #177777,CMPVIR          ;SET COMMAND VIRTUAL CIRCUIT ( 1 FOR DM)
003724  012767  177777  174342          MOV     #177777,RSPVIR          ;SET COMMAND VIRTUAL CIRCUIT
003732  000167  00032?                  JMP     SEND                    ;SEND THE PACKET
```

```
                                ;*****************************************************************************
                                ;                                                                            ;
                                ;       WRITE                                                                ;
                                ;                                                                            ;
                                ;       SET UP OP CODE, MODIFIERS,BUFFER SIZE (BYTE COUNT),                  ;
                                ;       BUFFER DESCRIPTOR (PYSICAL AND EXTENDED ADDRESS)                     ;
                                ;               LET READ SET SIMULAR DATA IN THE PACKET:                     ;
                                ;       DISK ADDRESS AND CYLINDER ID (LOGICAL BLOCK NUMBER),                 ;
                                ;       THEN SEND THE PACKET.                                                ;
                                ;                                                                            ;
                                ;*****************************************************************************
003736  004767  000712          WRITE:  JSR     PC,CLRPAK               ;CLEAR PACKETS
003742  012767  000042  174422          MOV     #OP.WR,P.OPCD+CMPACK    ;SET THE OPCODE
003750  016700  174166          WRITEA: MOV     WBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
003754  016767  174154  174420          MOV     WBUFPA,P.ADPA+CMPACK    ;SET THE BUFFER DESCRIPTOR(PA)
003762  016767  174464  174414          MOV     WBFFEA,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR(EA)
003770  000415                          BR      READA                   ;

                                ;*****************************************************************************
                                ;                                                                            ;
                                ;       READ                                                                 ;
                                ;                                                                            ;
                                ;       SET UP OP CODE, MODIFIERS,BUFFER SIZE (BYTE COUNT),                  ;
                                ;       BUFFER DESCRIPTOR (PYSICAL AND EXTENDED ADDRESS),                    ;
                                ;       DISK ADDRESS AND CYLINDER ID (LOGICAL BLOCK NUMBER),                 ;
                                ;       THEN SEND THE PACKET.                                                ;
                                ;                                                                            ;
                                ;*****************************************************************************
003772  004767  000656          READ:   JSR     PC,CLRPAK               ;CLEAR PACKETS
003776  012767  000041  174366          MOV     #OP.RD,P.OPCD.CMPACK    ;SET THE OPCODE
004004  016700  174122                  MOV     RBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
004010  016767  174434  174366          MOV     RBFFEA,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR
004016  016767  174104  174356          MOV     RBUFPA,P.ADPA+CMPACK    ;
004024  012767  000040  174240  READA:  MOV     #32.,RSPLEN             ;SET RESPONCE PACKET LENGTH
004032  012767  000040  174316          MOV     #32.,CMPLEN             ;SET COMMAND PACKET LENGTH
004040  006300                          ASL     R0                      ;MAKE IT NUMBER OF BYTES
004042  010067  174330                  MOV     R0,P.BCNT+CMPACK        ;SET READ BUFFER SIZE
004046  016767  174514  174342          MOV     SECL,P.LBN+CMPACK       ;SET LOGICAL BLOCK NUMBER
004054  016767  174510  174336          MOV     SECH,P.LBN+2+CMPACK     ;
004062  000476                          BR      SEND                    ;SEND THE PACKET
```

```
                                ;*****************************************************************************
                                ;
                                ;       DETERMINE ACCESS PATHS
                                ;
                                ;       SET UP CODE, GO SEND PACKET
                                ;
                                ;*****************************************************************************
004064  004767  000564          DAP:    JSR     PC,CLRPAK               ;CLEAR PACKETS
004070  012767  000013  174274          MOV     #OP.DAP,P.OPCD+CMPACK   ;SET OPCODE
004076  012767  000074  174166          MOV     #60.,RSPLEN             ;SET LENGTHS
004104  012767  000074  174244          MOV     #60.,CMPLEN             ;
004112  000462                          BR      SEND                    ;SEND THE PACKET

                                ;*****************************************************************************
                                ;
                                ;       AVAILABLE PACKET
                                ;
                                ;       SET OP CODE AND MODIFIERS THEN SEND THE PACKET
                                ;
                                ;*****************************************************************************
004114  004767  000534          AVAILB: JSR     PC,CLRPAK               ;CLEAR PACKETS
004120  012767  000010  174244          MOV     #OP.AVL,P.OPCD+CMPACK   ;SET THE OPCODE
004126  012767  000014  174136          MOV     #12.,RSPLEN             ;SET RESPONCE PACKET LENGTH
004134  000413                          BR      GTSTAA                  ;SEND THE PACKET

                                ;*****************************************************************************
                                ;
                                ;       GET UNIT STATUS
                                ;
                                ;       SET OPCODE AND MODIFIER (FOR THEN NEXT UNIT
                                ;               THEN SEND THE PACKET
                                ;
                                ;*****************************************************************************
004136  004767  000512          GTSTAT: JSR     PC,CLRPAK               ;CLEAR PACKETS
004142  012767  000003  174222          MOV     #OP.GUS,P.OPCD+CMPACK   ;SET THE OPCODE
004150  012767  000001  174216          MOV     #MD.NXU,P.MOD+CMPACK    ;CLEAR MODIFIERS
004156  012767  000060  174106          MOV     #48.,RSPLEN             ;SET RESPONCE PACKET LENGTH
004164  012767  000014  174164  GTSTAA: MOV     #12.,CMPLEN             ;SET COMMAND PACKET LENGTH
004172  000432                          BR      SEND                    ;SEND THE PACKET
```

```
 1                                      ;*****************************************************************************
 2                                      ;                                                                            ;
 3                                      ;       SET CONTROLLER CHARACTERISTICS
 4                                      ;
 5                                      ;       SET OP CODE AND CONTROLLER FLAG (ENABLE ATTENSION MSGS)
 6                                      ;       CLEAR MSCP VERSION, HOST TIMEOUT, USE FRACTION,
 7                                      ;               AND ALL OF QUAD WORD TIME AND DATE.
 8                                      ;       THEN SEND PACKET
 9                                      ;                                                                            ;
10                                      ;*****************************************************************************
11 004174                              SCC:
12 004174  004767  000454                       JSR     PC,CLRPAK               ;GO CLEAR THE COMMAND PACKET
13 004200  012767  000040  174150               MOV     #32.,CMPLEN             ;SET UP COMMAND PACKET LENGTH
14 004206  012767  000034  174056               MOV     #28.,RSPLEN             ;SET UP RESPONCE PACKET LENGTH
15 004214  012767  000004  174150               MOV     #OP.SCC,P.OPCD+CMPACK   ;SET THE OPCODE
16 004222  012767  000200  174150               MOV     #CF.AVL,P.CNTF+CMPACK   ;SET THE CONTROLLER FLAGS
17                                                                              ; TO ENABLE ATTENTION MSGS
18 004230  000413                               BR      SEND                    ;SEND THE PACKET
19
20                                      ;*****************************************************************************
21                                      ;                                                                            ;
22                                      ;       ONLINE
23                                      ;
24                                      ;       SET OPCODE, MODIFIERS, UNIT ID, HOST ID
25                                      ;       SHADOW UNIT, ERROR FLAGS
26                                      ;       THEN SEND PACKET
27                                      ;                                                                            ;
28                                      ;*****************************************************************************
29 004232  004767  000416              ONLINE: JSR     PC,CLRPAK               ;CLEAR PACKETS
30 004236  012767  00004C  174026               MOV     #32.,RSPLEN             ;SET RESPONCE PACKET LENGTH
31 004244  012767  000044  174104               MOV     #36.,CMPLEN             ;SET COMMAND PACKET LENGTH
32 004252  012767  000011  174112               MOV     #OP.ONL,P.OPCD+CMPACK   ;SET THE OPCODE
```

```
                                ;*****************************************************************************
                                ;                                                                            ;
                                ;       SEND - SEND A PACKET                                                 ;
                                ;       INTERP - WAIT FOR AN INTERRUPT                                       ;
                                ;                                                                            ;
                                ;       SET UP THE COMMAND REFERENCE NUMBER AND UNITNO IN THE PACKET          ;
                                ;       SET OWN, CLEAR FLAG IN THE COMMAND RING (FOR CONTROLLER)             ;
                                ;       SET OWN & FLAG IN MESSAGE RING (FOR INTERRUPTS BY CONTROLLER)        ;
                                ;       AFTER INTERRUPT, MAKE SURE THE PACKET WAS PROCESSED (NO HARD         ;
                                ;       OR SOFT ERRORS) THEN RETURN TO CYCLED.                               ;
                                ;                                                                            ;
                                ;       INPUT:  CMPACK IS FILLED EXCEPT FOR CMDREF & UNITNO                  ;
                                ;               INTERRUPT VECTOR AND BR LEVEL ARE ESTABLISHED                ;
                                ;                                                                            ;
                                ;       OUTPUT: MSPACK IS FILLED                                             ;
                                ;               CLEAR CARRY IF COMMAND PACKET WAS OK                         ;
                                ;                       ELSE GO DO A HARD/SOFT ERROR                         ;
                                ;                                                                            ;
                                ;*****************************************************************************
004260  005267  174004          SEND:   INC     CMDREF                  ;NEW COMMAND REFERENCE NUMBER
004264  001775                          BEQ     SEND                    ;COMMAND REF # CANNOT = 0
004266  016767  173776  174066          MOV     CMDREF,P.CRF+CMPACK     ;SET COMMAND REF NUMBER
004274  016767  174302  174064          MOV     UNITNO,P.UNIT+CMPACK    ;SET UNIT NUMBER
004302  042767  040000  173756          BIC     #RG.FLG,COMMND+2        ;CLEAR FLAG
004310  052767  100000  173750          BIS     #RG.OWN,COMMND+2        ;SET OWN FOR COMMAND RING
004316  052767  140000  173736          BIS     #<RG.OWN+RG.FLG>,RSPONC+2  ;SET OWN AND FLAG FOR MESSAGE RING
004324  005777  173456                  TST     @ADDR                   ;FORCE POLLING TO PACKET
004330                          INTERP:
004330  104400  000000'                  EXIT$.BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;
004334                          NTRUPT:
                                ;-------------------------------------------------------- -------
004334  000004  000000' 004342'         PIRQ$.BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                ;-------------------------------------------------- -- ---- ----- ---
                                ;
004342                          1$:
004342  005067  173710                  CLR     RINTR                   ;CLEAR INTERRUPT FLAG
004346  022767  000100  173732          CMP     #OP.AVA,P.OPCD+RSPACK   ;WAS AN AVAILABLE ATTENTION RECIEVED?
004354  001524                          BEQ     15$                     ;IF IT WAS, EXIT
004356  022767  000102  173722          CMP     #OP.ACP,P.OPCD+RSPACK   ;WAS THE ACCESS PATH ATTENTION RECIEVED?
004364  001527                          BEQ     16$                     ;IF IT WAS, GO PROCESS
                                                                        ; ELSE CHECK SUCCESS
004366                          2$:
004366  016700  173716                  MOV     P.STS+RSPACK,R0         ; SUCCESS?
004372  001513                          BEQ     14$                     ;IF YES, EXIT
004374  042700  177740                  BIC     #177740,R0              ;CLEAR UPPER 11 BITS OF SUB-STATUS
004400  001510                          BEQ     14$                     ;IF SUCCESS = 0, EXIT OK
004402  005067  173500                  CLR     ERRTYP                  ;IF GOT HERE, ERROR
004406  122700  000013                  CMPB    #ST.DRV,R0              ; DRIVE ERROR?
004412  001015                          BNE     3$                      ;IF NOT NEXT TEST
004414  032767  001000  173374          BIT     #SR.DUA,SR1             ;ARE WE DUAL PORTING?
004422  001472                          BEQ     12$                     ;IF NOT, GO REPORT ERROR/ELSE EXPECTED
004424  022767  000053  173656          CMP     #<ST.DRV+SC.STO>,P.STS+RSPACK   ;IS IT AN SDI RESPONCE TIMEOUT:
004432  001464                          BEQ     10$                     ;IF TRUE, DRIVE IS NOT ONLINE, EXIT
004434  022767  000113  173646          CMP     #<ST.DRV+SC.INV>,P.STS+RSPACK   ;IS IT THE INVALID SDI RESPONCE?
004442  001460                          BEQ     10$                     ;IF TRUE, DRIVE IS NOT ONLINE, EXIT
004444  000461                          BR      12$                     ;ELSE HARD ERROR
```

```
004446                                   3$:
004446  122700  000012                          CMPB    #ST.CNT,R0              ; CONTROLLER ERROR?
004452  001004                                  BNE     4$                      ;IF NOT NEXT TEST
004454  012767  000003  173424                  MOV     #ERR.3,ERRTYP           ;ELSE, SET ERROR TYPE
004462  000531                                  BR      ERRORH                  ;AND HARD ERROR
004464                                   4$:
004464  122700  000011                          CMPB    #ST.HST,R0              ; HOST BUFFER ACCESS ERROR?
004470  001004                                  BNE     5$                      ;IF NOT NEXT TEST
004472  012767  000032  173406                  MOV     #ERR.32,ERPTYP          ;ELSE, SET ERROR TYPE
004500  000522                                  BR      ERRORH                  ;AND HARD ERROR
004502                                   5$:
004502  122700  000010                          CMPB    #ST.DAT,R0              ; DATA ERROR?
004506  001004                                  BNE     6$                      ;IF NOT NEXT TEST
004510  012767  000001  173370                  MOV     #ERR.1,ERRTYP           ;ELSE, SET ERROR TYPE
004516  000533                                  BR      ERRORS                  ;AND SOFT ERROR
004520                                   6$:
004520  122700  000006                          CMPB    #ST.WPR,R0              ; WRITE PROTECTED?
004524  001431                                  BEQ     12$                     ;ELSE HARD ERROR
004526                                   8$:
004526  122700  000004                          CMPB    #ST.AVL,R0              ; STILL AVAILABLE?
004532  001005                                  BNE     9$                      ;IF NOT NEXT TEST
004534  022767  000003  173630                  CMP     #OP.GUS,P.OPCD+CMPACK   ;ELSE, IF COMMAND WAS
                                                                                ;        GET UNIT STATUS
004542  001427                                  BEQ     14$                     ;THEN EXPECTED & LEAVE ROUTINE
004544  000421                                  BR      12$                     ;ELSE HARD ERROR
004546                                   9$:
004546  122700  000003                          CMPB    #ST.OFL,R0              ; UNIT OFFLINE?
004552  001022                                  BNE     13$                     ;IF NOT NEXT TEST
                                         ; *** OFFLINE WHEN TRIED ONLINE OR GET UNIT STATUS
004554  022767  000011  173610                  CMP     #OP.ONL,P.OPCD+CMPACK   ;WAS IT AN ONLINE COMMAND?
004562  001410                                  BEQ     10$                     ;IF SO, SET CARRY/EXIT
004564  022767  000003  173600                  CMP     #OP.GUS,P.OPCD+CMPACK   ;IS IT GET UNIT STATUS COMMAND?
004572  001404                                  BEQ     10$                     ;IF SO, SET CARRY/EXIT
004574  022767  000042  173570                  CMP     #OP.WR,P.OPCD+CMPACK    ;IS IT WRITE COMMAND?
004602  001002                                  BNE     12$                     ;IF NOT, REPORT HARD ERROR
004604  000261                           10$:   SEC                             ;ELSE, SET CARRY TO
004606  000207                                  RTS     PC                      ;AND RETURN TO DROP DRIVE/AWAIT AVAILABLE DRIVE
                                         ; *** HARD ERROR EXIT WITH ERROR TYPE = 6
004610                                   12$:
004610  012767  000006  173270                  MOV     #ERR.6,ERRTYP           ;ELSE, SET ERROR TYPE
004616  000453                                  BR      ERRORH                  ;AND HARD ERROR
                                         ; *** SOFT ERROR EXIT WITH ERROR TYPE = 0
004620                                   13$:
004620  000472                                  BR      ERRORS                  ;ERROR WITH ERRTYP = 0 & IS A SOFT ERROR
                                                                                ;  ST.CMP,ST.MFE,.ST.ABO,ST.CMD

                                         ; *** SUCESSFUL EXIT
004622                                   14$:
004622  000241                                  CLC                             ;CLEAR CARRY 'CAUSE PACKET IS OK
004624  000207                                  RTS     PC                      ;ELSE, OK, SO FAR.
004626                                   15$:
                                         ; *** WAIT FOR ATTENTION INTERRUPT
                                         ; *** DID WE GET AN AVAILABLE ATTENTION MESSAGE THAT WE EXPECTED?
004626  005767  173632                          TST     EXPAV
004632  001004                                  BNE     16$                     ;IF EXPAV IS NOT 0, WE GOT ONE WE DIDN'T EXPECT
004634  012767  177777  173622                  MOV     #177777,EXPAV           ;CLEAR EXPECTED AVAILABLE ATTENTION MESSAGE WORD
004642  000767                                  BR      14$                     ; AND RETURN
004644                                   16$:
```

MODULE CODE

```
    113 004644  052767  140000  173410          BIS     @<RG.OWN+RG.FLG>,RSPONC+2       ;
    114 004652  000626                          BR      INTERP                          ;WAIT FOR RESPONCE OF LAST PACKET SENT
    115
```

```
                                ;*****************************************************************************
                                ;                                                                            *
                                ;       CLEAR PACKETS                                                        *
                                ;                                                                            *
                                ;       ASSUMPTION:  1)  RESPONCE BUFFER PRECEDES THE COMMAND BUFFER         *
                                ;                    2)  TWO WORDS BEFORE EACH BUFFER IS FOR LENGTH          *
                                ;                            OF PACKET AND VIRTUAL CIRCUIT                   *
                                ;                                                                            *
                                ;       OUTPUT: R2 = 0 WHEN DONE                                             *
                                ;               R5 = END OF COMMAND PACKET WHEN DONE                         *
                                ;                                                                            *
                                ;*****************************************************************************
004654                          CLRPAK:
004654  017767  173372  173572          MOV     @SAREG,NUM              ;IF SA REG NOT ZERO, STORE IN NUM
004662  001421                          BEQ     5$                      ;IF SA REG IS ZERO, CLEAR PACKETS
                                        ;*************************************************************
                                                                        ;CONVERT NUM TO ASCII AND
                                                                        ;STORE AT ADR1
004664  104420  000000' 000454'         OTOA$,BEGIN,NUM,ADR1
004672  000466'
                                        ;*************************************************************
004674  104403  000000' 005334'         MSGN$,BEGIN,SANOTO      ;ASCII MESSAGE CALL WITH COMMON HEADER
004702  010346                          MOV     R3,-(SP)                ;SAVE R3
004704  010446                          MOV     R4,-(SP)                ;SAVE R4
004706  004767  174532                  JSR     PC,INITUD               ;RE INIT SA REGISTER
004712  012603                          MOV     (SP)+,R3                ;RESTORE R3
004714  012604                          MOV     (SP)+,R4                ;RESTORE R4
004716  004767  177252                  JSR     PC,SCC                  ;SET CONTROLLER CHARS AGAIN
004722  005267  173116                  INC     HRDCNT                  ;INCREMENT HARD ERROR COUNT
                                                                        ;DOING THIS WILL CAUSE ANOTHER CALL TO CLRPAK
004726  012702  000064          5$:     MOV     #52.,R2                 ;R2 = # OF WORDS TO CLEAR
004732  012705  000272'                  MOV     #RSPLEN,R5              ;R5 -> RSPLEN, 1ST WORD TO CLEAR
004736  005025                  6$:     CLR     (R5)+                   ;CLEAR WORD
004740  005302                          DEC     R2                      ;R2 = ZERO? (DONE CONDITION)
004742  001375                          BNE     6$                      ;IF NOT ZERO, LOOP
004744  000207                          RTS     PC                      ;RETURN

                                ;*****************************************************************************
                                ;                                                                            *
                                ;       HARD ERROR      CARRY WILL BE SET                                    *
                                ;                                                                            *
                                ;*****************************************************************************
004746                          ERRORH:
004746  032767  000004  173042          BIT     #SR.REP,SR1             ;DO WE REPORT THE ERROR?
004754  001403                          BEQ     7$                      ;IF SO, REPORT
004756  005267  173062                  INC     HRDCNT                  ;ELSE, INCREMENT THE HARD ERROR
                                                                        ; COUNT IF NOT REPORTED
004762  000407                          BR      8$                      ;SKIP REPORT
004764  004767  000056          7$:     JSR     PC,SETTAB               ;SET UP TABLE
                                        ;*************************************************************
004770  104405  000000' 000000          HRDER$,BEGIN,NULL       ;
                                        ;*************************************************************
004776  004767  000070                  JSR     PC,PRINTE
005002  000261                  8$:     SEC
005004  000207                          RTS     PC                      ;RETURN TO CYCLED

                                ;*****************************************************************************
                                ;                                                                            *
```

```
                                ;       SOFT ERROR      CARRY WILL BE SET                                       ;
                                ;                                                                               ;
                                ;*******************************************************************************
                                ERRORS:
005006                                  BIT     @SR.REP,SR1             ;DO WE REPORT THE ERROR?
005006  032767  000004  173002          BEQ     9$                      ;IF SO, REPORT
005014  001403                          INC     SOFCNT                  ;ELSE, INCREMENT THE HARD ERROR
005016  005267  173020                                                  ; COUNT IF NOT REPORTED
                                        BR      10$                     ;SKIP REPORT
005022  000407                  9$:     JSR     PC,SETTAB               ;SET UP TABLE
005024  004767  000016                  ;*******************************************************************
                                        SOFER$,BEGIN,NULL       ;
005030  104406  000000' 000000          ;*******************************************************************
                                        JSR     PC,PRINTE
005036  004767  000030          10$:    SEC                             ;SET CARRY
005042  000261                          RTS     PC                      ;RETURN TO CYCLED
005044  000207
                                ;*******************************************************************************
                                ;                                                                               ;
                                ;       SETTAB                                                                  ;
                                ;                                                                               ;
                                ;       SET UP A TABLE OF VALUES FOR A SOFT OR HARD ERROR                       ;
                                ;                                                                               ;
                                ;*******************************************************************************
                                SETTAB:
005046                                  MOV     ADDR,CSRA               ;SET UP CONTROL STATUS REG REPORT
005046  016767  172734  173024          MOV     P.STS+RSPACK,ASTAT      ;SET UP STATUS
005054  016767  173230  173022          MOV     @SAREG,ACSR             ;REPORT WHAT IS STATUS REG
005062  017767  173164  173012          RTS     PC
005070  000207
                                ;*******************************************************************************
                                ;
                                ;       PRINT EXTENDED ERROR MESSAGE
                                ;
                                ;       PRINT STATUS, OPCODE, UNIT NUMBER, BYTE COUNT, LBN AND ADDRESS
                                ;
                                ;*******************************************************************************
005072                          PRINTE:
                                        ;***************************************************************
                                                                ;CONVERT P.STS+RSPACK TO ASCII AND
                                                                ;STORE AT ADR1
005072  104420  000000' 000310'          OTOA$,BEGIN,P.STS+RSPACK,ADR1
005100  000466'
                                        ;***************************************************************
                                        ;***************************************************************
                                                                ;CONVERT P.OPCD+RSPACK TO ASCII AND
                                                                ;STORE AT ADR2
005102  104420  000000' 000306'          OTOA$,BEGIN,P.OPCD+RSPACK,ADR2
005110  000475'
                                        ;***************************************************************
                                        ;***************************************************************
                                                                ;CONVERT P.UNIT+RSPACK TO ASCII AND
                                                                ;STORE AT ADR3
005112  104420  000000' 000302'          OTOA$,BEGIN,P.UNIT+RSPACK,ADR3
005120  000504'
                                        ;***************************************************************
                                        ;***************************************************************
                                                                ;CONVERT P.BCNT+RSPACK TO ASCII AND
```

```
                                                        ;STORE AT ADR4
    005122  104420  000000' 000312'          OTOA$,BEGIN,P.BCNT*RSPACK,ADR4
    005130  000513'
                                                        ;*****************************************************************
                                                        ;*****************************************************************
90                                                      ;CONVERT P.LBN*2*CMPACK TO ASCII AND
                                                        ;STORE AT ADR5
    005132  104420  000000' 000420'          OTOA$,BEGIN,P.LBN*2*CMPACK,ADR5
    005140  000522'
                                                        ;*****************************************************************
                                                        ;*****************************************************************
91                                                      ;CONVERT P.LBN*CMPACK TO ASCII AND
                                                        ;STORE AT ADR6
    005142  104420  000000' 000416'          OTOA$,BEGIN,P.LBN*CMPACK,ADR6
    005150  000531'
                                                        ;*****************************************************************
                                                        ;*****************************************************************
92                                                      ;CONVERT P.ADEA*CMPACK TO ASCII AND
                                                        ;STORE AT ADR7
    005152  104420  000000' 000404'          OTOA$,BEGIN,P.ADEA*CMPACK,ADR7
    005160  000540'
                                                        ;*****************************************************************
                                                        ;*****************************************************************
93                                                      ;CONVERT P.ADPA*CMPACK TO ASCII AND
                                                        ;STORE AT ADR8
    005162  104420  000000' 000402'          OTOA$,BEGIN,P.ADPA*CMPACK,ADR8
    005170  000547'
                                                        ;*****************************************************************
94  005172  104403  000000' 005376'          MSGN$,BEGIN,BANNER      ;ASCII MESSAGE CALL WITH COMMON HEADER
95  005200  000207                           RTS     PC
96
```

```
                                        .SBTTL  MODULE MESSAGES
005202  005444'                 INITE1: MSG2
005204  005515'                         MSG4
005206  177777                          177777

005210  005475                  INITER: MSG3
005212  000466'                         ADR1
005214  005675'                         MSG10
005216  000475'                         ADR2
005220  006023'                         MSG14
005222  000504'                         ADR3
005224  177777                          177777

005226  005444'                 INITE2: MSG2
005230  005542'                         MSG5
005232  177777                          177777

005234  005444'                 INITE3: MSG2
005236  005560'                         MSG6
005240  177777                          177777

005242  005652'                 DRP1:   MSG8
005244  000475'                         ADR2
005246  005662'                         MSG9
005250  006245'                         MSG20
005252  000466'                         ADR1
005254  006732'                         MSGD1
005256  177777                          177777

005260  005652'                 DRP2:   MSG8
005262  000475                          ADR2
005264  005662'                         MSG9
005266  006245'                         MSG20
005270  000466'                         ADR1
005272  006776'                         MSGD2
005274  177777                          177777

005276  005652'                 DRP3:   MSG8
005300  000475'                         ADR2
005302  005662'                         MSG9
005304  006245'                         MSG20
005306  000466'                         ADR1
005310  007040'                         MSGD3
005312  177777                          177777

005314  005707'                 ERRPAS: MSG11
005316  000475'                         ADR2
005320  005733'                         MSG12
005322  000504'                         ADR3
005324  005771'                         MSG13
005326  000466'                         ADR1
005330  005442'                         MSG1
005332  177777                          177777

005334  006105'                 SANOTO: MSG17
005336  000466'                         ADR1
005340  006142'                         MSG18
```

```
005342  177777                   177777

005344  006034'        UNIOFF:   MSG16
005346  000466'                  ADR1
005350  177777                   177777

005352  006556'        WARN1:    MSG40
005354  177777                   177777

005356  006433'        WARN2:    MSG37
005360  177777                   177777

005362  006364'        WARN3:    MSG36
005364  177777                   177777

005366  005615'        ABORT:    MSG7
005370  177777                   177777

005372  006215'        ZERO:     MSG19
005374  177777                   177777

005376  006270'        BANNER:   MSG21
005400  000466'                  ADR1
005402  006362'                  MSG23
005404  000475'                  ADR2
005406  006362'                  MSG23
005410  000504'                  ADR3
005412  006362'                  MSG23
005414  000513'                  ADR4
005416  006362'                  MSG23
005420  000522'                  ADR5
005422  006362'                  MSG23
005424  000531'                  ADR6
005426  006362'                  MSG23
005430  000540'                  ADR7
005432  006362'                  MSG23
005434  000547'                  ADR8
005436  005442'                  MSG1
005440  177777                   177777
```

```
                                        .SBTTL  MORE MODULE MESSAGES
                                        .NLIST  BEX

005442    045    000           MSG1:    .ASCIZ  '%'
005444    045    103    117    MSG2:    .ASCIZ  '%CONTROLLER INIT ERROR, '
005475    045    123    101    MSG3:    .ASCIZ  '%SA REGISTER = '
005515    106    117    125    MSG4:    .ASCIZ  'FOUND BY DIAGNOSTIC '
005542    123    124    105    MSG5:    .ASCIZ  'STEP NOT SET.'
005560    105    130    120    MSG6:    .ASCIZ  'EXPECTED DATA WAS INCORRECT '
005615    045    122    105    MSG7:    .ASCIZ  '%RETRY COUNT EXCEEDED, ABORT'
005652    045    104    122    MSG8:    .ASCIZ  '%DRIVE '
005662    040    104    122    MSG9:    .ASCIZ  ' DROPPED. '
005675    040    111    116    MSG10:   .ASCIZ  ' IN STEP '
005707    045    123    117    MSG11:   .ASCIZ  '%SOFT ERROR COUNT #'
005733    040    040    040    MSG12:   .ASCIZ  '        *** HARD ERROR COUNT #'
005771    045    103    110    MSG13:   .ASCIZ  '%CHECK DATA ERROR COUNT #'
006023    045    101    104    MSG14:   .ASCIZ  '%ADDR = '
006034    045    125    116    MSG16:   .ASCIZ  '%UNIT WAS FOUND OFFLINE.  UNIT NUMBER = '
006105    045    123    101    MSG17:   .ASCIZ  '%SA REGISTER IS NOT ZERO, = '
006142    045    103    117    MSG18:   .ASCIZ  '%CONTROLLER IS GOING THROUGH INITILIZATION'
006215    045    122    111    MSG19:   .ASCIZ  '%RING AREA NOT CLEARED%'
006245    045    104    105    MSG20:   .ASCIZ  '%DEVICE ID BIT =  '
006270    045    123    124    MSG21:   .ASCIZ  '%STATUS ENDCOD UNITNU BYTECO HI LBN LO LBN EXTADR PHYADR%'
006362    040    000           MSG23:   .ASCIZ  ' '
006364    040    041    040    MSG36:   .ASCIZ  ' ! OPERATING WITH NO DISK ACCESSING !%'
006433    007    007    040    MSG37:   .ASCII  <07><07>' ! CUSTOMER DATA WILL BE OVERWRITTEN !%'
006504    040    055    055             .ASCIZ  '                    ------------------------------------%'<07><07>
006556    040    111    106    MSG40:   .ASCII  ' IF YOU WISH TO DESTROY CUSTOMER DATA, SET BIT1 (NOT BIT0)%'
006651    040    111    116             .ASCIZ  ' IN SWITCH REGISTER 1(SR1) OF DUBD? EQUAL TO 1.%'
006732    045    105    122    MSGD1:   .ASCIZ  '%ERRORS CAUSED DRIVE TO BE DROPPED%'
006776    045    125    116    MSGD2:   .ASCIZ  '%UNIT WAS NOT FOUND BY EXERCISER%'
007040    045    104    126    MSGD3:   .ASCIZ  '%DVID1 BIT SET HIGHER THAN ACTUAL # OF DRIVES FOUND%'
                                        .EVEN
007126                         RBUF:    .BLKW   256.            ;THE READ BUFFER
      000001                            .END
```

Symbol table

```
ABORT   005366R
ACSR    000102R
ADDR    000006R
ADDR22= 001000
ADR1    000466R
ADR2    000475R
ADR3    000504R
ADR4    000513R
ADR5    000522R
ADR6    000531R
ADR7    000540R
ADR8    000547R
ASB     000106R
ASR04   002034R
ASTAT   000104R
AVAILB  004114R
AWAS    000110R
BANNER  005376R
BEGIN   000000R
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BREAK$= 104407
BR1     000012R
BR2     000013R
BTOD$ = 104421
CDATA$= 104412
CDERCT  000144R
CDWDCT  000146R
CF.AVL= 000200
CF.MSC= 000100
CF.OTH= 000040
CF.SHD= 000002
CF.THS= 000020
CF.576= 000001
CINTR   000254R
CLRPAK  004654R
CMDREF  000270R
CMPACK  000362R
CMPLEN  000356R
CMPVIR  000360R
COMMND  000264R
CONFIG  000056R
CSRA    000100R
CYCLED  002624R
CYCLEL  003122R
DAP     004064R
DATCK$= 104411
DATER$= 104404
DOINTR  003104R
DROP1   003410R
DROP2   003420R
DROP3   003430R
DROP4   003436R
DRP1    005242R
DRP2    005260R
DRP3    005276R
DVICE   000600R
DVID1   000014R
EA      000446R
EA22    000564R
EF.BBR= 000200
EF.BBU= 000100
EF.FRS= 000200
EF.LOG= 000040
EF.LST= 000100
EF.MIS= 000001
EF.SEX= 000020
ENDIT$= 104413
END$  = 104410
ERRORH  004746R
ERRORS  005006R
ERROR1  002150R
ERROR2  002146R
ERROR3  002144R
ERROR5  002132R
ERRPAS  005314R
ERRTYP  000106R
ERR.0 = 000000
ERR.1 = 000001
ERR.3 = 000003
ERR.32= 000032
ERR.6 = 000006
EXIT$ = 104400
EXPAV   000464R
FREE    000150R
GETPA$= 104415
GTSTAA  004164R
GTSTAT  004136R
GWBUF$= 104414
HC.CCT= 000006
HC.CMD= 000004
HC.CPK= 000356R
HC.RCT= 000002
HC.RES= 000000
HC.RPK= 000276R
HC.SIZ= 000010
HRDCNT  000044R
HRDER$= 104405
HRDPAS  000050R
ICONT   000036R
ICOUNT  000040R
IDNUM   000122R
IMODX.= 000000
INIT    000030R
INITER  005210R
INITE1  005202R
INITE2  005226R
INITE3  005234R
INITUD  001444R
INTA    002072R
INTERP  004330R
INTR    000120R
LIMIT   000576R
LOOP1   001300R
LOOP2   001330R
L.CHVR= 000015
L.CNTI= 000014
L.CYL = 000034
L.DATA= 000050
L.ERLC= 000030
L.EVNT= 000000
L.GRP = 000040
L.SCTR= 000042
L.SLOT= 000002
L.TRCK= 000041
L.UHVR= 000027
L.UNTI= 000016
L.USVR= 000026
L.VSER= 000044
MAITP   003666R
MAITR   003562R
MAITW   003616R
MAP22$= 104416
MA10NC  001412R
MD.CMP= 040000
MD.ERR= 010000
MD.EXP= 100000
MD.FEU= 000001
MD.NXU= 000001
MD.SCH= 004000
MD.SCL= 002000
MD.SEC= 001000
MD.SER= 000400
MD.SPD= 000001
MD.SSH= 000200
MD.VOL= 000002
MD.WBN= 000100
MD.WBV= 000040
MODNAM  000C00R
MODSP   000252R
MSGD1   006732R
MSGD2   006776R
MSGD3   007040R
MSGN$ = 104403
MSGS$ = 104402
MSG$  = 104401
MSG1    005442R
MSG10   005675R
MSG11   005707R
MSG12   005733R
MSG13   005771R
MSG14   006023R
MSG16   006034R
MSG17   006105R
MSG18   006142R
MSG19   006215R
MSG2    005444R
MSG20   006245R
MSG21   006270R
MSG23   006362R
MSG3    005475R
MSG36   006364R
MSG37   006433R
MSG4    005515R
MSG40   006556R
MSG5    005542R
MSG6    005560R
MSG7    005615R
MSG8    005652R
MSG9    005662R
NTRUPT  004334R
NULL  = 000000
NUM     000454R
OLDEA   000460R
OLDPA   000456R
ONEFIL= 000001
ONLINE  004232R
OPEN  = 000000
OP.ABO= 000001
OP.ACC= 000020
OP.ACP= 000102
OP.AVA= 000100
OP.AVL= 000010
OP.CCD= 000021
OP.CMP= 000040
OP.DAP= 000013
OP.END= 000200
OP.ERL= 000101
OP.ERS= 000022
OP.FLU= 000023
OP.GCS= 000002
OP.GUS= 000003
OP.MRD= 000030
OP.MWR= 000031
OP.ONL= 000011
OP.RD = 000041
OP.RPL= 000024
OP.SCC= 000004
OP.SMC= 000102
OP.SUC= 000012
OP.WR = 000042
OTOA$ = 104420
PA      000444R
PASCNT  000034R
PA18    000556R
PA22    000562R
PICKBK  003154R
PIRQ$ = 000004
PKTSIZ= 000060
POPSP = 005726
POPSP2= 022626
PORTID  000606R
PRINTE  005072R
PRNMSG  000462R
PRTNUM= 000017
PRTY  = 000000
PRTY0 = 000000
PRTY1 = 000040
PRTY2 = 000100
PRTY3 = 000140
PRTY4 = 000200
PRTY5 = 000240
PRTY6 = 000300
PRTY7 = 000340
PS    = 177776
PSW   = 177776
PUSH  = 005746
PUSH2 = 024646
PWRFLG= 000002
P.ADEA= 000022
P.ADPA= 000020
P.BCNT= 000014
P.BUFF= 000020
P.CMST= 000020
P.CNCL= 000022
P.CNT = 000006
P.CNTF= 000016
P.CNTI= 000024
P.CPSP= 000042
P.CRF = 000000
P.CTMO= 000020
P.CYL = 000050
P.ELGF= 000034
P.FBBK= 000034
P.FLGS= 000011
P.GRP = 000046
P.HSTI= 000020
P.HTMO= 000020
P.LBN = 000034
P.LGDT= 000014
P.MEDI= 000034
```

Symbol table

```
P.MLUN= 000014          RESTRT  001036R         SA.S1 = 004000          SR4     000024R         UF.CMW= 000002
P.MOD = 000012          RESTR1  001116R         SA.S2 = 010000          START   000660R         UF.INA= 040000
P.OPCD= 000010          RESTR2  001072R         SA.S3 = 020000          STAT    000026R         UF.RMV= 000200
P.OTRF= 000014          RES1    000056R         SA.S4 = 040000          ST.ABO= 000002          UF.RPL= 010000
P.RBN = 000014          RES2    000060R         SA.VCE= 000177          ST.AVL= 000004          UF.SCH= 004000
P.RBNS= 000056          RG.FLG= 040000          SA.VEC= 000177          ST.CMD= 000001          UF.SCL= 002000
P.RCTC= 000057          RG.OWN= 100000          SBADR   000102R         ST.CMP= 000007          UF.WBN= 000040
P.RCTS= 000054          RINTR   000256R         SCC     004174R         ST.CNT= 000012          UF.WPH= 020000
P.RGID= 000034          RLIM  = 000004          SC.DIS= 000400          ST.DAT= 000010          UF.WPS= 010000
P.RGOF= 000040          RSPACK  000276R         SC.DUP= 000200          ST.DIA= 000037          UF.576= 000004
P.SFTW= 000040          RSPLEN  000272R         SC.INV= 000100          ST.DRV= 000013          UNIOFF  005344R
P.SHST= 000042          RSPONC  000260R         SC.IOP= 000100          ST.HST= 000011          UNITFL  000610R
P.SHUN= 000040          RSPVIR  000274R         SC.NVL= 000040          ST.MFE= 000005          UNITNO  000602R
P.STS = 000012          RSTRT   000112R         SC.STO= 000040          ST.MSK= 000037          UNSZH   000574R
P.SZOF= 000012          R6      =%000006        SECH    000570R         ST.OFL= 000003          UNSZL   000572R
P.TIME= 000024          R7      =%000007        SECL    000566R         ST.SUB= 000040          VA      000442R
P.TRCK= 000044          SANOTO  005334R         SEND    004260R         ST.SUC= 000000          VECTOR  000010R
P.UNFL= 000016          SAREG   000252R         SEQACC  003342R         ST.WPR= 000006          WARN1   005352R
P.UNIT= 000004          SA.CMD= 034000          SETTAB  005046R         SVR0    000062R         WARN2   005356R
P.UNSZ= 000044          SA.CME= 000360          SETUP   002320R         SVR1    000064R         WARN3   005362R
P.UNTI= 000024          SA.DIA= 000400          SNDSTP  002046R         SVR2    000066R         WASADR  000104R
P.USEF= 000022          SA.ERC= 003777          SOFCNT  000042R         SVR3    000070R         WBFFEA  000452R
P.VRSN= 000014          SA.ERR= 100000          SOFER$= 104406          SVR4    000072R         WBUFEA  000136R
P.VSER= 000050          SA.GO = 000001          SOFPAS  000046R         SVR5    000074R         WBUFPA  000134R
RANACC  003164R         SA.INE= 000200          SPOINT  000032R         SVR6    000076R         WBUFRQ  000140R
RAND$ = 10441           SA.INT= 000200          SPSIZ = 000040          SYSCNT  000052R         WBUFSZ  000142R
RANNUM  000054R         SA.LFC= 040000          SR.CMP= 004000          TABLEW  000614R         WDFR    000116R
RBFFEA  000450R         SA.MAP= 000100          SR.DUA= 001000          TEND    000654R         WDTO    000114R
RBUF    007126R         SA.MCV= 000377          SR.REP= 000004          TIMER = 002260          WORK    000612R
RBUFEA  000130R         SA.NSI= 002000          SR.SEQ= 002000          TIMOUT= 005670          WRITE   003736R
RBUFPA  000126R         SA.PRG= 000001          SR.SUM= 000010          TRPDFD= 000022          WRITEA  003750R
RBUFSZ  000132R         SA.Q22= 001000          SR.XFR= 000002          TRY     000604R         XFLAG   000005R
RBUFVA  000124R         SA.RSE= 000017          SR1     000016R         TSTOFL  002554R         XMEM    000560R
READ    003772R         SA.RSP= 003400          SR2     000020R         UF.CMR= 000001          ZERO    005372R
READA   004024R         SA.SM = 000040          SR3     000022R

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        010126    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12704 Words  ( 50 Pages)
Size of core pool: 14336 Words  ( 56 Pages)
Operating  system: RT-11 (Under RTEM 11)

Elapsed time: 00:00:54.00
XDUBDO,XDUBDO/C=XDUBDO.DOC,DDXCOM.MAC,XDUBDO.MAC
```

Cross reference table (CREF V05.01)

ABORT 17-55 28-730
ACSR 4-30 27-76*
ADDR 4 30 14-68 16-26* 17-48 26-27 27-74
ADDR22 4-30
ADR1 4-61@ 15-34 15-35* 17-34 21-33 27-15 27-86 28-7 28-26 28-34 28-42 28-51 28 56 28-61
28-80
ADR2 4-63@ 15-30 15-31* 17-32 21-31 21-32* 27-87 28-9 28-23 28-31 28-39 28-47 28-82
ADR3 4-65@ 15-32 15-33* 17-48 27-88 28-11 28-49 28-84
ADR4 4-67@ 27-89 28-86
ADR5 4-69@ 27-90 28-88
ADR6 4-71@ 27-91 28-90
ADR7 4-73@ 27-92 28-92
ADR8 4-75@ 27-93 28-94
ASB 4-30
ASR04 15-41 15-59 15-90 16-64 16-82 16-89 16-109@
ASTAT 4-30 27-75*
AVAILB 19-89 24-21@
AWAS 4-30
BANNER 27-94 28-79@
BEGIN 4-30 14-50 14-53 14-56 14-65 15-15 15-30 15-32 15-34 15-36 15-39 15-55 15-57 15-80
15-88 15-93 16-23 16-31 16-31 16-79 16-86 16-132 16-135 17-25 17-26 17-32 17-34 17-38
17-42 17-46 17-48 17-49 17-50 17-55 17-56 19-85 19-115 20-15 20-17 21-31 21-33 21-37
21-41 21-43 26-29 26-32 27-15 27-16 27-44 27-61 27-86 27-87 27-88 27-89 27-90 27-91
27-92 27-93 27-94
BIT0 4-30 7-57
BIT00 6-30
BIT01 4-15 6-40
BIT02 4-16 6-50
BIT03 4-17 6-60
BIT04 6-70
BIT05 6-80
BIT06 6-90
BIT07 6-10@ 16-51 16-53
BIT08 6-11@
BIT09 4-18 6-120
BIT1 4-30
BIT10 4-30 4-19 6-130
BIT11 4-30 4-20 6-140
BIT12 4-30 6-150
BIT13 4-30 6-160
BIT14 4-30 6-170 8-4
BIT15 4-30 6-180 8-3 16-45 16-50
BIT2 4-30
BIT3 4-30
BIT4 4-30
BIT5 4-30
BIT6 4-30
BIT7 4-30
BIT8 4-30
BIT9 4-30
BR1 4-30 16-129
BR2 4-30
BREAK$ 4-30 16-31 16-31
BTOD$ 4 30 15-30 15-32 15-34 21-31
CDATA$ 4-30 19-85 19-115
CDERCT 4-30 14-58* 15-34

Cross reference table (CREF V05.01)

```
CDHDCT    4-3#
CF.S76   11-8#
CF.AVL   11-3#     25-16
CF.MSC   11-4#
CF.OTH   11-5#
CF.SHD   11-7#
CF.THS   11-6#
CINTR     4-25#
CLRPAK   22-13     22-33     23-12     23-30     24-8      24-21     24-34     25-12     25-29     27-12#
CMDREF    4-31#    14-64*    16-70     18-19*    26-20*    26-22
CMPACK    4-39#    16-78     22-14*    22-15*    22-16*    22-34*    22-35*    22-36*    22-43*    22-46*    23-13*    23-15*    23-16*    23-31*
          23-33*   23-34*    23-38*    23-39*    23-40*    24-9*     24-22*    24-35*    24-36*    25-15*    25-16*    25-32*    26-22*    26-23*
          26-77    26-85     26-87     26-89     27-90     27-91     27-92     27-93
CMPLEN    4-37#    22-45*    23-36*    24-11*    24-38*    25-13*    25-31*
CMPVIR    4-38#    22-47*
COMMND    4-29#     4-41     16-80*    16-83*    26-24*    26-25*
CONFIG    4-3#
CSRA      4-3#     27-74*
CYCLED   15-69     19-18#    19-48
CYCLEL   15-92     19-110#
DAP      24-8#
DATCK$    4-3#
DATER$    4-3#
DOINTR   19-97#
DROP1    15-71     21-16#
DROP2    18-34     21-19#
DROP3    18-52     21-22#
DROP4    21-18     21-21     21-24#
DRP1     21-37     28-22#
DRP2     21-41     28-30#
DRP3     21-43     28-38#
DVICE     5-18#    14-61*    15-52     15-64     15-77     18-61     21-24     21-30*
DVID1     4-3#     14-61
EA        4-43#    15-18     15-21     16-63     16-81     16-88
EA22      5-8#
EF.BBR   10-20#
EF.BBU   10-21#
EF.FRS   13-3#
EF.LOG   10-22#
EF.LST   13-4#
EF.MIS   13-5#
EF.SEX   10-23#
END$      4-3#     15-55     17-26     17-56
ENDIT$    4-3#     15-80     15-93
ERR.0     6-23#
ERR.1     6-24#    26-69
ERR.3     6-25#    26-59
ERR.32    6-27#    26-64
ERR.6     6-26#    26-95
ERROR1   16-36     16-139    17-30#
ERROR2   16-144    17-29#
ERROR3   16-40     16-54     16-60     17-28#
ERROR5   16-69     17-25#
ERRORH   18-90     19-77     26-60     26-65     26-96     27-37#
ERRORS   26-70     26-99     27-54#
ERRPAS   15-36     28-46#
```

Cross reference table (CREF V05.01)

```
ERRTYP    4-30    26-46*  26-59*  26-64*  26-69*  26-95*
EXIT$     4-39    16-132  26-29
EXPAV     4-580   14-59*  19-35*  19-98*  26-108  26-110*
FREE      4-30
GETPA$    4-30    15-15   15-39   16-23   16-79   16-86
GTSTAA   24-24    24-380
GTSTAT   18-30    19-30   19-68   24-340
GWBUF$    4-30    15-57   15-88
HC.CCT    8-140
HC.CMD    8-130
HC.CPK    8-160
HC.RCT    8-120
HC.RES    8-110
HC.RPK    8-150    8-16
HC.SIZ    8-80
HRDCNT    4-30    15-32   27-23*  27-40*
HRDER$    4-30    17-49   27-44
HRDPAS    4-30
ICONT     4-30
ICOUNT    4-30
IDNUM     4-30
IMODX.    4-30    15-57   15-88
INIT      4-30
INITE1   17-38    28-20
INITE2   17-42    28-140
INITE3   17-46    28-180
INITER   17-50    28-60
INITUD   15-22    16-220   27-19
INTA     16-128   16-1340
INTERP   19-37    19-100   26-280   26-114
INTR      4-30
L.CHVR   13-130
L.CNTI   13-110   13-120
L.CYL    13-180
L.DATA   13-230
L.ERLC   13-170
L.EVNT   13-90
L.GRP    13-190
L.SCTR   13-210
L.SLOT   13-100
L.TRCK   13 200
L.UHVR   13-160
L.UNTI   13-140
L.USVR   13-150
L.VSER   13-220
LIMIT     5-160   22-37   22-39
LOOP1    15-53    15-570  15-81
LOOP2    15-630   15-78
MA10NC   15-45    15-880  15-94
MAITP    22-18    22-40   22-420
MAITR    19-112   22-130
MAITW    19-111   22-330
MAP22$    4-30
MD.CMP   10 30
MD.ERR   10 50
MD EXP   10 40
```

Cross reference table (CREF V05.01)

```
MD.FEU  10-14#
MD.NXU  10-16#   24-36
MD.SCH  10-6#
MD.SCL  10-7#
MD.SEC  10-8#
MD.SER  10-9#
MD.SPD  10-13#
MD.SSH  10-10#
MD.VOL  10-15#
MD.WBN  10-11#
MD.WBV  10-12#
MODNAM   4-3#
MODSP    4-3      4-3#
MSG$     4-3#
MSG1    28-52    28-95    29-4#
MSG10   28-8     29-13#
MSG11   28-46    29-14#
MSG12   28-48    29-15#
MSG13   28-50    29-16#
MSG14   28-10    29-17#
MSG16   28-60    29-18#
MSG17   28-55    29-19#
MSG18   28-57    29-20#
MSG19   28-76    29-21#
MSG2    28-2     28-14    28-18    29-5#
MSG20   28-25    28-33    28-41    29-22#
MSG21   28-79    29-23#
MSG23   28-81    28-83    28-85    28-87    28-89    28-91    28-93    29-24#
MSG3    28-6     29-6#
MSG36   28-70    29-25#
MSG37   28-67    29-26#
MSG4    28-3     29-7#
MSG40   28-64    29-28#
MSG5    28-15    29-8#
MSG6    28-19    29-9#
MSG7    28-73    29-10#
MSG8    28-22    28-30    28-38    29-11#
MSG9    28-24    28-32    28-40    29-12#
MSGD1   28-27    29-30#
MSGD2   28-35    29-31#
MSGD3   28-43    29-32#
MSGN$    4-3#    14-50    14-53    14-56    15-36    17-25    17-38    17-42    17-46    17-50    17-55    21-37    21-41    21-43
        27-16    27-94
MSGS$    4-3#
NTRUPT  15-37    16-91    26-31#
NULL     4-3#    17-49    27-44    27-61
NUM      4-48#   17-31*   17-32    17-33*   17-34    21-17*   21-20*   21-23*   21-35*   21-39*   27-13*   27-15
OLDEA    4-50#   14-71*   15-18    15-21*
OLDPA    4-49#   14-70*   15-16    15-20*
ONEFIL   1-4#     1-8      1-581    2-1
ONLINE  19-42    25-29#
OP.ABO   9-3#
OP.ACC   9-4#
OP.ACP   9-25#   26-38
OP.AVA   9-22#   26-36
OP.AVL   9-5#    24-22
```

Cross reference table (CREF V05.01)

```
OP.CCD    9-6#
OP.CMP    9 7#
OP.DAP    9 8#      24-9
OP.END    9-21#
OP.ERL    9 23#
OP.ERS    9 9#
OP.FLU    9-10#
OP.GCS    9-11#
OP.GUS    9-12#     24-35     26-77     26-87
OP.MRD    9-19#     22-14
OP.MWR    9-20#     22-34
OP.ONL    9-13#     25-32     26-85
OP.RD     9-14#     23-31
OP.RPL    9-15#
OP.SCC    9-16#     25-15
OP.SHC    9-24#
OP.SUC    9-17#
OP.WR     9-18#     23-13     26-89
OPEN      4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3
          4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3
          4-3#      4-42      4-43
OTOA$     4-3#      17-32     17-34     17-48     21-33     27-15     27-86     27-87     27-88     27-89     27-90     27-91     27-92     27-93
P.ADEA    11-20#    22-15*    22-35*    23-16*    23-33*    27-92
P.ADPA    11-19#    22-16*    22-36*    23-15*    23-34*    27-93
P.BCNT    11-17#    12-9#     22-43*    23-38*    27-89
P.BUFF    11-18#
P.CMST    12-15#
P.CNCL    12-44#
P.CNT     12-53#
P.CNTF    11-40#    12-42#    25-16*
P.CNTI    12-45#
P.CPSP    11-33#
P.CRF     11-13#    12-4#     12-51#    26-22*
P.CTMO    12-43#
P.CYL     12-26#
P.ELGF    11-31#
P.FBBK    12-10#
P.FLGS    12-7#     12-55#
P.GRP     12-25#
P.MSTI    11-29#    12-20#    12-34#
P.MTMO    11-41#
P.LBN     11-21#    23-39*    23-40*    27-90     27-91
P.LGDT    12-57#
P.MEDI    12-46#
P.MLUN    12-18#    12-32#
P.MOD     11-16#    24-36*
P.OPCD    11-15#    12-6#     12-54#    22-14*    22-34*    23-13*    23-31*    24-9*     24-22*    24-35*    25-15*    25-32*    26-36     26-38
          26-77     26-85     26-87     26-89     27-87
P.OTRF    11-25#    12-14#
P.RBN     11-36#
P.RBNS    12-28#
P.RCTC    12-29#
P.RCTS    12-27#
P.RGID    11-46#    22-46*
P.RGOF    11-47#
P.SFTW    11-22#    12-11#
```

Cross reference table (CREF V05.01)

```
P.SHST  12-23#  12-47#
P.SHUN  11 32#  12-22#  12-36#
P.STS   12-8#   18-80   18-84   26-42   26-51   26-53   27-75   27-86
P.SZOF  12-56#
P.TIME  11 43#
P.TRCK  12-24#
P.UNFL  11-28#  12-19#  12-33#
P.UNIT  11-14#  12-5#   12-52#  18-36   26-23*  27-88
P.UNSZ  12-37#  19-44   19-45
P.UNTI  11-30#  12-21#  12-35#
P.USEF  11-42#
P.VRSN  11-39#  12-41#
P.VSER  12-38#
PA       4-42#  15-16   15-20   16-55   16-80   16-87
PA18     5-5#
PA22     5-7#
PASCNT   4-3#   15-27
PICKBK  19-67   20-11#
PIRQ$    4-3#   16-135  26-32
PKTSIZ   8-9#    8-16
POPSP    4-3#
POPSP2   4-3#
PORTID   5-21#  15-66*  18-25*  18-51*  21-24   21-26   21-30   21-33
PRINTE  27-45   27-62   27-85#
PRNMSG   4-54#  14-60*  15-27   15-29*
PRTNUM   4-53#   4-54   14-60   15-29
PRTY     4-3#
PRTY0    4-3     4-3#
PRTY1    4-3#
PRTY2    4-3#
PRTY3    4-3#
PRTY4    4-3      4-3#
PRTY5    4-3#
PRTY6    4-3#
PRTY7    4-3#
PS       4-3#
PSW      4-3#
PUSH     4-3#
PUSH2    4-3#
PWRFLG   4-3#
R6       4-3#
R7       4-3#
RANACC  20-14#
RAND$    4-3#   14-65   20-15   20-17
RANNUM   4-3#   14-66   20-16   20-18
RBFFEA   4-45#  15-42*  22-15   23-33
RBUF     4-3    29-34#
RBUFEA   4-3#   15-40
RBUFPA   4-3#   19-85   19-115  22-16   23-34
RBUFSZ   4-3#   22-17   23-32
RBUFVA   4-3#   15-39
READ    19-80   23-30#
READA   23-17   23-35#
RES1     4-3#
RES2     4-3#
RESTR1  15-19   15-24#
```

Cross reference table (CREF V05.01)

```
RESTR2  15-17    15-20#
RESTRT   4-3     15-13#
RG.FLG   8-4#    19-36    19-99    26-24    26-26    26-113
RG.OWN   8-3#    19-36    19-99    26-25    26 26    26-113
RINTR    4-26#   26-35*
RLIM     5-26#   17-52
RSPACK   4-35#    8-15    16-85    18-36    18-80    18-84    19-44    19-45    26-36    26-38    26-42    26-51    26-53    27-75
        27-86    27-87    27-88    27-89
RSPLEN   4-33#   22-44*   23-35*   24-10*   24-23*   24-37*   25-14*   25-30*   27-26
RSPONC   4-28#   15-14    16-22    16-66    16-87*   16-90*   19-36*   19-99*   26-26*   26-113*
RSPVIR   4-34#   22-48*
RSTRT    4-3#
SA.CMD   7-29#
SA.CME   7-44#
SA.DIA   7-20#   16-38
SA.ERC   7-13#
SA.ERR   7-9#    16-29    16-138
SA.GO    7-57#   16-76
SA.INE   7-34#
SA.INT   7-27#   16-45
SA.LFC   7-49#
SA.MAP   7-21#   16-37
SA.MCV   7-53#
SA.NSI   7-18#
SA.PRG   7-39#
SA.Q22   7-19#   16-37
SA.RSE   7-43#
SA.RSP   7-28#
SA.S1    7-5#    16-35    16-38    16-48
SA.S2    7-6#    16-53
SA.S3    7-7#
SA.S4    7-8#
SA.SM    7-22#   16 37
SA.VCE   7-33#
SA.VEC   7-26#
SANOTO  27-16    28-55#
SAREG    4-23#   14-68*   14-69*   16-28    16-77*   16-130*  16-137   17-33    27-13    27-76
SBADR    4-3#
SC.DIS  13-49#   18-80    18-84
SC.DUP  13-51#   18-80    18-84
SC.INV  13-56#   26-53
SC.IOP  13-48#   18-80    18-84
SC.NVL  13-46#   18-80    18-84
SC.STO  13-55#   26-51
SCC     18-18    25-11#   27-22
SECH     5-11#   14-67*   20-24*   20-40*   20-41    20-44*   20-49*   20-70*   20-72    20-75*   23-40
SECL     5-10#   14-66*   20-57*   20-64*   20-66    20-74*   23-39
SEND    22-49    23-41    24-12    24-39    25-18    26-20#   26-21
SEQACC  20 13    20-63#
SETTAB  27-43    27-60    27-73#
SETUP   15-51    18-16#
SNDSTP  16-49    16-56    16-65    16-127#
SOFCNT   4-3#    15-30    27-57*
SOFERR   4-3#    27-61
SOFPAS   4-3#
SPOINT   4-3#
```

Cross reference table (CREF V05.01)

```
SPSIZ     3 230     4-3
SR.CMP    4 200     19-82     19-113
SR.DUA    4 180     19-19     19-75     19-86     26-49
SR.REP    4 160     27-38     27-55
SR.SEQ    4 190     20-12
SR.SUM    4 170     15-25
SR.XFR    4 150     14 49     14-51     15-44
SR1       4 30      14-49*    14-51     15-25     15-44     19-19     19-75     19-82     19-86     19-113    20-12     26-49     27-38     27-55
SR2       4 30      16-72
SR3       4-30
SR4       4-30
ST.ABO   13 310
ST.AVL   13-330     19-70     26-75
ST.CMD   13-300
ST.CMP   13-360
ST.CNT   13-390     26-57
ST.DAT   13-370     26-67
ST.DIA   13-410
ST.DRV   13-400     18-78     26-47     26-51     26-53
ST.HST   13-380     26-62
ST.MFE   13-340
ST.MSK   13-270     18-84
ST.OFL   13-320     18-76     26-82
ST.SUB   13-280
ST.SUC   13-290
ST.WPR   13-350     26-72
START     4 3       14-460    17-54
STAT      4-30
SVR0      4-30
SVR1      4-30
SVR2      4-30
SVR3      4-30
SVR4      4-30
SVR5      4-30
SVR6      4-30
SYSCNT    4-30
TABLEW    5-280     14-62*    14-63*    15-61     18-22     18-39
TEND      5-310     15-75     18-41     18-65
TIMER     4-560     16-27
TIMOUT    5-250
TRPDFD    4-3       4 3       4-3       4-3       4-3       4-3       4-3       4 3       4-3       4-3       4-3       4 3       4-3       4 3
          4-3       4-3       4-3       4-3       4-3       4-3       4-3       4-3       4 3       4-3       4 3       4-3       4-3       4-3
          4-3       4-3       4-3       4-3       4-3       4-3       4-3       4 3       4 30      4 30      4 30      4-30      4-30      4 30
          4 30      4-30      4-30      4-30      4-30      4-30      4-30      4-30      4 30      4 30      4 30      4 30      4 30
TRY       5-200     15-23*    16-92*    17-51*    17 52     19-22     19-50*
TSTOFL   18 760     19-32
UF.576   10-380
UF.CMR   10-280
UF.CMW   10-290
UF.INA   10-310
UF.RMV   10 320
UF.RPL   10 300
UF.SCH   10-330
UF.SCL   10 340
UF.WBN   10 350
UF.WPH   10 360
```

Cross reference table (CREF V05.01)

```
LF.WPS    10 3*0
UNIOFF    28 60#
UNITFL     5 22#
UNITNO     5 19#    15 43#  15-67#  18-23#  18 24   18 36#  18 45   18-50#  18-56   18 58   18 63#  21 31   26 23
UNSZM      5 14#    19-44#  19-47   20-25   20-28   20 41   20 72
UNSZL      5 13#    19-45#  19-65#  20 50   20-52   20-54   20 66
VA         4-41#    15 14#  15-15   16-22#  16 23   16 78#  16 79   16 85#  16-86
VECTOR     4 3#     15-37#  16-42   16-91#  16-127
WARN1     14-50     28-64#
WARN2     14-53     28-67#
WARN3     14-56     28-70#
WASADR     4 3#
WBFFEA     4-46#    15-60#  15-91#  22-35   23-16
WBUFEA     4-3#     15-58   15-89
WBUFPA     4-3#     22-36   23-15
WBUFRQ     4-3#
WBUFSZ     4-3#     19-60   22-37   22-41   23-14
WDFR       4-3#
WDTO       4-3#
WORK       5-23#    18-29#  18-32#
WRITE     19-73     23-12#
WRITEA    23-14#
XFLAG      4-3#
XMEM       5-6#
ZERO      17-25     28-76#
```

DUBD DEC/X11 SYSTEM EXERCISER M MACRO V05.01b Thursday 20 Sep 84 10:17 Page M 1
Cross reference table (CREF V05.01)

```
BKMOD     3 1110
BREAK     3 2010  16-31
BTOD      3 2200  15-30   15-32   15-34   21-31
CKDATA    3-2560  19-85   19-115
DATACK    3-2650
DATERR    3 1540
DFSEVN    3-2880   4 3     4-3     4-3     4-3     4 3     4-3     4-3     4-3     4-3     4-3     4-3     4-3
DSEVNT    3-2980   4-3     4-3     4-3     4-3     4-3     4 3
END       3-1910  17-26
ENDIT     3-1820  15-80   15-93
ENDMOD    3-1870  15-55   17-56
EQUATS    3-3060   4-3
EXIT      3-1360  16-132  26-29
GETPA     3-2470  15-15   15-39   16-23   16-79   16-86
GMBUFF    3-2350  15-57   15-88
HRDER     3-1440  17-49   27-44
IOMOD     3-1070
IOMODP    3-1310
IOMODR    3-1270
IOMODX    3-1230   4-3
MAP22     3-2510
MODULE    3-240    4-3
MSG       3-1700
MSGN      3-1740  14-50   14-53   14-56   15-36   17-25   17-38   17-42   17-46   17-50   17-55   21-37   21-41   21-43
         27-16    27-94
MSGS      3-1780
NBKMOD    3-1190
OTOA      3-2060  17-32   17-34   17-48   21-33   27-15   27-86   27-87   27-88   27-89   27-90   27-91   27-92   27-93
PIRQ      3-1950  16-135  26-32
RAND      3-1400  14-65   20-15   20-17
SBKMOD    3-1150
$DFER     3-1600  27-61
```