Micro Fiche Scan

Name of device(s) tested:
    RA60/80/81,UDA50,KDA50
Test description:
    UDA50/KDA50 DEC/X11 MOD
MAINDEC Number or Package Identifier (after SEP 1977):
    CXDUBE0
Fiche Document Part Number:
    AH-S915E-MC
Fiche preparation date unknown, using copyright year:
    1985
Image resolution:
    8-bit gray levels, max. quality for archiving

COPYRIGHT (C) 1981-85 by d|i|g|i|t|a|l

IDENTIFICATION

PRODUCT CODE: AC-S914C-MC

PRODUCT NAME: CXDUBE0 - UDA50A/KDA50Q DEC/X11 MOD

PRODUCT DATE: 1-OCT-1985

MAINTAINER: ROGER OAKEY

AUTHOR: MATT TEDONE

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DEC	DIBOL	RSX
DEC/CMS	EduSystem	UNIBUS
DECnet	IAS	VAX
DECsystem-10	MASSBUS	VMS
DECSYSTEM-20	PDP	VT
DECUS	PDT	Digital Logo
DECwriter	RSTS	

```
.ENABL LC

.REM &
```

TABLE OF CONTENTS

1.0 ABSTRACT

The exerciser will be similar to that of other disk subsystem exercisers. Writes will be performed to the disks followed by read and compare of the data read. The controller will do all error retrying. Errors will be reported on the console terminal.

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, other PDP-11 diagnostic programs are available. They are:

CZUDHA1 - UDA50-A/KDA50-Q Basic Subsystem Diagnostic
CZUDIA0 - UDA50-A/KDA50-Q Disk Drive Exerciser
CZUDJA0 - UDA50-A/KDA50-Q Disk Subsystem Exerciser
CZUDKA0 - UDA50-A/KDA50-Q Disk Formatter.

2.0 REQUIREMENTS

Hardware for all cases:
One DEC/X11 module configures for one UDA50-A 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.

Memory: DUBE 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.

```
DUBE0 PA:0060162      APC: 000674     PASS #00000

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

DUBE0 PA:0060210      APC: 000722     PASS #00000
'! 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 CXDUBE0 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.

```
DUBE0 PA:0060210      APC: 000722     PASS #00000
'! OPERATING WITH NO DISK ACCESSING !'
```

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

```
DUBE0 PA:0060210      APC: 000722     PASS #00000
'! 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 DUBE 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 DUBE 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 bus 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 characteristics.
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.

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 desirable.

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

```
DUBE0 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:

```
DUBE0 PA: 00064116 APC: 004630 PASS: 00000 ERRCNT: 00001
CSRA: 172150 CSRC: 000000 ASTAT: 000006 ERRTYP: 000006
RUNTIME: 000:00:22

DUBE0 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:

```
DUBE0 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:

```
DUBE0 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:

```
DUBE0 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/ Timeout test error
    105102 - D PROC no board 2 error/D PROC control reg test error/
             D PROC RAM parity error
    105105 - D proc RAM buffer error
    105152 - D proc SDI error
    105153 - D proc write mode wrap SERDES 16 error
    105154 - D PROC read mode, SERDES 16, 10 RSGEN and
             ECC circuitry error
    106040 - U proc ALU error/DFAIL test error/
             Unexpected trap error
    106041 - U proc Control Register error
    106042 - U PROC parity error set erroneously/
             CROM parity test error
    106047 - U proc Constant ROM error with D proc running SDI test
    106055 - Unexpectant trap found, aborted diagnostic
    106071 - U PROC Log/Antilog RAM checksum error
    106072 - U PROC ROM parity test 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.

 DUBE0 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 controller did not successfully clear the ring buffer in
the host area, the following message will be printed.

 DUBE0 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 SA register displays a non-zero value after the initialization sequence is done, the followng message will be printed.

```
DUBE0 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 into the IP register.
100001 - bus envelope/packet read error (parity or timeout)
100002 - bus envelope/packet write error (parity or timeout)
100003 - controller ROM and RAM parity error
100004 - controller RAM parity error
100005 - controller ROM parity error
100006 - Ring read error (parity or timeout)
100007 - Ring write error (parity of timeout)
100010 - bus interrupt master failure
100011 - Host access timeout error
100012 - Host exceeded credit limit
100013 - Q-bus master error
100014 - Diagnostic controller fatal error
100015 - Hardware timeout of instruction loop
100016 - Invalid virtual circuit identifier
100017 - Interrupt write error on bus
100020 - Maintenance read/write invalid region identifier
100021 - Maintenance write load to non-loadable controller
100022 - Controller RAM error (non-parity)
100023 - INIT sequence error
100024 - High-level protocol incompatibility error
100025 - Purge/poll hardware failure
100026 - Mapping register read error (parity or timeout)
100027 - Mapping option unsupported
```

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:

```
DUBE0 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:

```
DUBE0 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:

```
DUBE0 PA: 00063012 APC: 003524 PASS #00000

DRIVE 00000 DROPPED.
DEVICE ID BIT = 000001
DVID1 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

DUBE 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.
---E	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 and 22 bit addresses from 16 bit addresses.

13.0 BIBLIOGRAPHY

CXQUADO 'DEC/X11 USER'S MANUAL' Sept 1984

CXQUBGO 'DEC/X11 CROSS-REFERENCE MANUAL' Sept 1982

CXQUCAO 'DEC/X11 REFERENCE CARD' January 1979

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

```
                                        &

588                     .DSABL LC
```

```
                                .SBTTL  MODULE HEADER BLOCK
                                        IOMODX <DUBE  >,172150,154,4,0,0,1000,104,RBUF,256.,256.
                                        MODULE  150000,DUBE  ,172150,154,4,0,0,1000,104,RBUF,256.,256.
                                        .TITLE  DUBE DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6.4     28-JAN-82
                                        .LIST   BIN
                                ;*****************************************************************************
                                BEGIN:
000000                          MODNAM: .ASCII  /DUBE / ;MODULE NAME.
000000  104  125  102
000003  105  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  000710'                 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  001066'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114  000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116  000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
```

MODULE HEADER BLOCK

```
000120  000000          INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122  000104          IDNUM:  104                     ;MODULE IDENTIFICATION NUMBER=104
000124  007616'          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     DON'T TEST STEP 4 COMPLETION.
                        ;                       DON'T 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.
                        ;       VERSION 4.0     JFM - 27-SEP-85
                        ;                       22-BIT Q-BUS ADDRESSING SUPPORT ADDED.
                        ;                       COMMENTS CLEANED UP AND UNUSED CODE DELETED.
                        ;                       DOCUMENTATION HAS BEEN UPDATED SOMEWHAT.

        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  ;DISK ACCESS MODE 0 = SEQUENTIAL, 1 = RANDOM
        004000          SR.CMP = BIT11  ;NO DATA COMPARE 0 = DO DATA COMPARE, 1 = DON'T DO DATA COMPARE

000252  000000          SAREG:  .WORD   0               ; CONTROLLER STATUS REG
                        ;++
                        ; THE ORDER OF THE NEXT 5 VARIABLES MUST NOT CHANGE
                        ;
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          RSPPA:  .WORD   0       ; RESPONSE RING
000274  000000          RSPEA:  .WORD   0       ;   PHYSICAL
000276  000000          RSPPP:  .WORD   0       ;     ADDRESS
000300  000000          RSPEP:  .WORD   0       ;       STORAGE
                                ;
000302  000000          RSPLEN: .WORD   0       ;RESPONCE PACKET LENGTH
000304  000000          RSPVIR: .WORD   0       ;RESPONCE PACKET VIRTUAL CIRCUIT
000306                  RSPACK: .BLKW   24.     ;RESPONCE PACKET
000366  000000          RPAKPA: .WORD   0       ; RESPOINSE PACKET
000370  000000          RPAKEA: .WORD   0       ;   PHYSICAL
000372  000000          RPAKPP: .WORD   0       ;     ADDRESS
000374  000000          RPAKEP: .WORD   0       ;       STORAGE
                                ;
000376  000000          CMPLEN: .WORD   0       ;COMMAND PACKET LENGTH
000400  000000          CMPVIR: .WORD   0       ;COMMAND PACKET VIRTUAL CIRCUIT
000402                  CMPACK: .BLKW   24.     ;COMMAND PACKET
000462  000000          CPAKPA: .WORD   0       ; COMMAND PACKET
000464  000000          CPAKEA: .WORD   0       ;   PHYSICAL
000466  000000          CPAKPP: .WORD   0       ;     ADDRESS
000470  000000          CPAKEP: .WORD   0       ;       STORAGE
                                ;
000472  000000          VA:     .WORD   0       ;GENERIC VIRTUAL ADDRESS FOR GETPA
000474  000000          PA:     .WORD   0       ;GENERIC PHYSICAL ADDRESS
000476  000000          EA:     .WORD   0       ;GENERIC EXTENDED ADDRESS
000500  000000          PA22:   .WORD   0       ; 22-BIT PHYSICAL ADDRESS
000502  000000          EA22:   .WORD   0       ; EE-BIT EXTENDED ADDRESS
                                ;
000504  000000          RBUFPP: .WORD   0       ;READ  BUFFER PHYSICAL ADDRESS SAVE AREA
000506  000000          RBUFEP: .WORD   0       ;READ  BUFFER EXTENDED ADDRESS SAVE AREA
000510  000000          WBUFPP: .WORD   0       ;WRITE BUFFER PHYSICAL ADDRESS SAVE AREA
000512  000000          WBUFEP: .WORD   0       ;WRITE BUFFER EXTENDED ADDRESS SAVE AREA
                                ;
000514  000000          NUM:    .WORD   0       ;ADDRESS USED IN OTOA
000516  000000          OLDPA:  .WORD   0       ;THE OLD PHYSICAL ADDRESS
000520  000000          OLDEA:  .WORD   0       ;THE OLD EXTENDED ADDRESS TO CHECK IF
                                ;                       CONTROLLER WILL BE REINITED
                                ;
                                ;
        000017          PRTNUM = 15.            ;PRINT MESSAGE EVERY 15TH TIME
000522  000017          PRNMSG: .WORD   PRTNUM  ;PRINT WORD SAVES THE VALUE TO CHECK FOR THE
                                ;                NEXT TIME AN END OF PASS MESSAGE IS WRITTEN
        002260          TIMER = 1200.           ;TIME TO WAIT 2-3 SECONDS AFTER DAP COMMAND
                                ;
000524  177777          EXPAV:  .WORD   177777  ;EXPECTING AN AVAILABLE ATTENTION MESSAGE = 0
                                ;                ELSE = 177777
                                ;
000526                  ADR1:   .BLKB   6
000534  000                             .BYTE   0
000535                  ADR2:   .BLKB   6
000543  000                             .BYTE   0
000544                  ADR3:   .BLKB   6
000552  000                             .BYTE   0
000553                  ADR4:   .BLKB   6
000561  000                             .BYTE   0
000562                  ADR5:   .BLKB   6
```

MODULE STORAGE AREA

```
     94 000570      000                     .BYTE   0
     95 000571                      ADR6:   .BLKB   6
     96 000577      000                     .BYTE   0
     97 000600                      ADR7:   .BLKB   6
     98 000606      000                     .BYTE   0
     99 000607                      ADR8:   .BLKB   6
    100 000615      000                     .BYTE   0
    101                                     .EVEN
```

```
                                .SBTTL MORE MODULE STORAGE

000616  000000                  SECL:   .WORD   0               ;CURRENT SECTOR LO ORDER ADDRESS
000620  000000                  SECH:   .WORD   0               ;CURRENT SECTOR HI ORDER ADDRESS

000622  000000                  UNSZL:  .WORD   0               ;UNIT SIZE LO ORDER LIMIT FROM ONLINE CMND
000624  000000                  UNSZH:  .WORD   0               ;UNIT SIZE HI ORDER LIMIT

000626  003300                  LIMIT:  .WORD   3300            ;4K - 1200 = MOST WORDS MAITW CAN TAKE

000630  000001                  DVICE:  .WORD   1               ;DEVICE TO TEST
000632  000000                  UNITNO: .WORD   0               ;UNIT NUMBER
000634  000000                  TRY:    .WORD   0               ;NUMBER OF TRIES
000636  000001                  PORTID: .WORD   1               ;BIT POSITION SELECTS THE PORT
000640  000000                  UNITFL: .WORD   0               ;SAVE UNIT FLAGS
000642  000000                  WORK:   .WORD   0               ;TEMPORARY WORK AREA

        005670                  TIMOUT = 3000.                  ;TIME OUT GUADGE
        C00004                  RLIM = 4                        ;RETRY LIMIT

000644  000000  000001          TABLEW: .WORD   0,1             ;TABLE ENTRY UNITNO,PORTID
000650  177777  177777                  .WORD   -1,-1           ;CURRENT LAST TABLE ENTRY
000654                                  .BLKW   12.             ;REST OF TABLE
000704  177777  177777          TEND:   .WORD   -1,-1           ;END MARKER
```

```
                                .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 EXISTENT
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

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 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
```

CONTROLLER BIT DEFINITIONS

```
    213         000001                  SA.GO=  BIT0                    ;GO BIT TO START CONTROLLER FIRMWARE
```

```
                                .SBTTL COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS

                                RG.OWN= BIT15                           ;SET WHEN CONTROLLER OWNS RING
100000                          RG.FLG= BIT14                           ;FLAG BIT

                                ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING

                                HC.SIZ= 8.                              ;SIZE OF HOST COMM AREA IN BYTES
000010                          PKTSIZ= 48.                             ;SIZE OF PACKETS IN BYTES
000060

                                HC.RES= 0.                              ;RESPONCE RING START
000000                          HC.RCT= 2.                              ;RESPONCE RING CONTROL WORD
000002                          HC.CMD= 4.                              ;COMMAND RING START
000004                          HC.CCT= 6.                              ;CONTROL RING CONTROL WORD
000006                          HC.RPK= RSPACK                          ;START OF RESPONCE PACKET BUFFER
000306'                         HC.CPK= HC.RPK+PKTSIZ                   ;START OF COMMAND PACKET BUFFER
000366'
```

```
                                .SBTTL COMMAND PACKET OPCODES

000001                          OP.ABO= 01                      ;ABORT COMMAND
000020                          OP.ACC= 20                      ;ACCESS COMMAND
000010                          OP.AVL= 10                      ;AVAILABLE COMMAND
000021                          OP.CCD= 21                      ;COMPARE CONTROLLER DATA COMMAND
000040                          OP.CMP= 40                      ;COMPARE HOST DATA COMMAND
000013                          OP.DAP= 13                      ;DETERMINE ACCESS PATHS COMMAND
000022                          OP.ERS= 22                      ;ERASE COMMAND
000023                          OP.FLU= 23                      ;FLUSH COMMAND
000002                          OP.GCS= 02                      ;GET COMMAND STATUS COMMAND
000003                          OP.GUS= 03                      ;GET UNIT STATUS COMMAND
000011                          OP.ONL= 11                      ;ONLINE COMMAND
000041                          OP.RD=  41                      ;READ COMMAND
000024                          OP.RPL= 24                      ;REPLACE COMMAND
000004                          OP.SCC= 04                      ;SET CONTROLLER CHARACTERISTICS COMMAND
000012                          OP.SUC= 12                      ;SET UNIT CHARACTERISTICS COMMAND
000042                          OP.WR=  42                      ;WRITE COMMAND
000030                          OP.MRD= 30                      ;MAINTENANCE READ COMMAND
000031                          OP.MWR= 31                      ;MAINTENANCE WRITE COMMAND
000200                          OP.END= 200                     ;END PACKET FLAG
000100                          OP.AVA= 100                     ;AVAILABLE ATTENTION MESSAGE
000101                          OP.ERL= 101                     ;ERROR LOG ATTENTION MESSAGE
000102                          OP.SHC= 102                     ;SHADOW COPY COMPLETE ATTENION MESSAGE
000102                          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

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

                                .SBTTL END PACKET FLAGS

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

                                .SBTTL UNIT FLAGS

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

END PACKET OFFSETS

```
    403         000012                  P.SZOF= 10.                     ;SIZE OR OFFSET
    404         000014                  P.LGDT= 12.                     ;START OF ERROR LOG DATA
```

ERROR LOG FLAGS

```
                        .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
```

ERROR LOG FLAGS

```
    461         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.                                    ;
                        ;                                                                            ;
                        ;*****************************************************************************
                        START:
000710                          INC     #-1                     ;FIRST TIME THRU HERE?
000710  005227  177777          BNE     1$                      ;BR IF NO
000714  001006                  BIC     #SR.XFR,SR1             ;DO NOT ALLOW DISK TRANSFERS
000716  042767  000002  177072  MSGN$,BEGIN,WARN1       ;ASCII MESSAGE CALL WITH COMMON HEADER
000724  104403  000000' 006042'
000732  032767  000002  177056  1$:     BIT     #SR.XFR,SR1             ;WILL CUSTOMER DATA BE OVERWRITTEN?
000740  001404                  BEQ     2$                      ;BR IF NO
000742  104403  000000' 006046' MSGN$,BEGIN,WARN2       ;ASCII MESSAGE CALL WITH COMMON HEADER
000750  000403                  BR      3$                      ;
000752                          2$:
```

MODULE CODE

```
000752  104403  000000' 006052'          MSGN$,BEGIN,WARN3       ;ASCII MESSAGE CALL WITH COMMON HEADER
000760                            3$:
000760  005067  177160                    CLR     CDERCT                  ;CLEAR DATA CHECK ERROR COUNT
000764  012767  177777  177532            MOV     #177777,EXPAV           ;NOT EXPECTING AN INTERRUPT
000772  012767  000017  177522            MOV     #PRTNUM,PRNMSG          ;INITIALIZE PRINT WORD
001000  016767  177010  177622            MOV     DVID1,DVICE             ;DVICE HAS DESIRED BITS SET
001006  005067  177632                    CLR     TABLEW                  ;SET TABLE FOR UNIT 0
001012  012767  000001  177626            MOV     #1,TABLEW+2             ;SET TABLE FOR PORTID FOR UNIT 0
001020  005067  177244                    CLR     CMDREF                  ;COMMAND REF # = 0
001024  104417  000000'                  RAND$,BEGIN
001030  016767  177020  177560            MOV     RANNUM,SECL
001036  005067  177556                    CLR     SECH                    ; FOR RESTARTING (INITIAL SECTOR ADDR)
001042  016767  176740  177202            MOV     ADDR,SAREG              ; STORE IN SA REG
001050  062767  000002  177174            ADD     #2,SAREG                ; SA REGISTER HAS PROPER ADDRESS
001056  005067  177434                    CLR     OLDPA                   ; OLD PHYSICAL ADDRESS CLEARED
001062  005067  177432                    CLR     OLDEA                   ; OLD EXTENDED ADDRESS CLEARED
                                                                          ; 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.                                               ;
                                ;                                                                            ;
                                ;*****************************************************************************
001066                          RESTRT:
001066  004767  000740                JSR     PC,CVTADR               ;
001072  026767  177200  177416        CMP     RSPPP,OLDPA             ;IS THE OLD PHYS ADDR = NEW ONE?
001100  001004                        BNE     RESTR2                  ;IF SO, REINIT
001102  026767  177172  177410        CMP     RSPEP,OLDEA             ;IS THE OLD EXTN ADDR = NEW ONE?
001110  001412                        BEQ     RESTR1                  ;IF NOT, DON'T REINIT
001112  C16767  177160  177376  RESTR2: MOV    RSPPP,OLDPA             ;ELSE SET THE OLD RING ADDR
001120  016767  177154  177372        MOV     RSPEP,OLDEA             ; AND THE OLD EXTENED ADDR
001126  004767  000260                JSR     PC,INITUD               ;AND INIT THE CONTROLLER
001132  005067  177476                CLR     TRY                     ;CLEAR RETRY COUNT
001136                          RESTR1:
001136  032767  000010  176652        BIT     @SR.SUM,SR1             ;DO WE WANT THE REPORT?
001144  001034                        BNE     1$                      ;IF NOT, SKIP THE REPORT
001146  026767  177350  176660        CMP     PRNMSG,PASCNT           ;DO WE PRINT?
001154  001030                        BNE     1$                      ;IF PASS COUNT IS NOT = PRINT WORD, SKIP
001156  062767  000017  177336        ADD     #PRTNUM,PRNMSG          ;PRINT WORD IS INCREMENT
                                ;*****************************************************************************
                                                                ;CONVERT SOFCNT TO ASCII AND
                                                                ;STORE AT ADR2
001164  104421  000000' 000042'        BTOD$,BEGIN,SOFCNT,ADR2
001172  000535'
                                ;*****************************************************************************
001174  105067  177342                CLRB    ADR2+5
                                ;*****************************************************************************
                                                                ;CONVERT HRDCNT TO ASCII AND
                                                                ;STORE AT ADR3
001200  104421  000000' 000044'        BTOD$,BEGIN,HRDCNT,ADR3
001206  000544'
                                ;*****************************************************************************
001210  105067  177335                CLRB    ADR3+5
                                ;*****************************************************************************
                                                                ;CONVERT CDERCT TO ASCII AND
                                                                ;STORE AT ADR1
001214  104421  000000' 000144'        BTOD$,BEGIN,CDERCT,ADR1
001222  000526'
                                ;*****************************************************************************
001224  105067  177303                CLRB    ADR1+5
001230  104403  000000' 006004'        MSGN$,BEGIN,ERRPAS      ;ASCII MESSAGE CALL WITH COMMON HEADER
001236  012777  005024' 176544  1$:    MOV     @NTRUPT,@VECTOR         ;GET VECTOR ADDRESS
                                                                ;SET POINTER
001244  005067  177362                CLR     UNITNO                  ;PRESET UNIT #
001250  032767  000002  176540        BIT     #SR.XFR,SR1             ;DISK XFER???
001256  001446                        BEQ     MA10NC                  ;NO!  DO MAINTENENCE (DISKLESS) ROUTINES
```

MODULE CODE

```
                                ;*****************************************************************************
                                ;       DO THE DISK OPERATIONS                                                ;
                                ;            CHECK TO SEE WHICH PORTS ARE AVAILABLE                           ;
                                ;*****************************************************************************

001260  004767  001524                JSR     PC,SETUP                ;FIND DRIVES/SET UP TABLE
001264  005767  177340                TST     DVICE                   ;ELSE, TEST FOR ANY MORE DRIVES
001270  001002                        BNE     LOOP1                   ;IF TRUE, DO A CYCLE

001272  104410  000000'               END$,BEGIN              ;

001276  004767  000420        LOOP1:  JSR     PC,GETWB                ; ALLOCATE WRITE BUFFER
001302  012704  000644'                MOV     #TABLEW,R4              ;R4 -> TABLE OF UNITNO AND PORTID
001306  012703  000001                MOV     #1,R3                   ;R3 IS AN INDEX TO DVICE
001312                          LOOP2:
001312  030367  177312                BIT     R3,DVICE                ;HAS THE DRIVE BEEN DROPPED
001316  001412                        BEQ     9$                      ;IF SO, SKIP THIS DRIVE
001320  016467  000002  177310        MOV     2(R4),PORTID            ;SET UP PORTID
001326  011467  177300                MOV     (R4),UNITNO             ;SET UP UNITNO
                                ; *** DO A DISK CYCLE
001332  004767  001756                JSR     PC,CYCLED               ;DO A CYCLE FOR DISK OPERATION
001336  103002                        BCC     9$                      ;IF SUCCESSFUL, CONTINUE
001340  004767  002534                JSR     PC,DROP1                ;IF NOT, DROP DRIVE
001344                          9$:
001344  062704  000004                ADD     #4,R4                   ;POINT TO NEXT ENTRY OF THE TABLE
001350  006303                        ASL     R3                      ;R3 POINTS TO NEXT BIT
001352  022704  000704'               CMP     #TEND,R4                ; POINT BEYOND LAST ENTRY?
001356  001403                        BEQ     12$                     ; IF NOT, THEN TRY AGAIN.
001360  020367  177244                CMP     R3,DVICE                ;IF R3 > DVICE THEN DONE WITH ITERATION
001364  003752                        BLE     LOOP2                   ;IF < OR =, LOOP
001366                          12$:
001366  104413  000000'               ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
001372  000741                        BR      LOOP1                   ;AND DO AGAIN


                                ;*****************************************************************************
                                ;       MAINTENENCE ROUTINE, DO THE DISKLESS CODE                             ;
                                ;*****************************************************************************

001374  004767  000322        MA1ONC: JSR     PC,GETWB                        ; GET WRITE BUFFER
001400  004767  002206                JSR     PC,CYCLEL
001404  104413  000000'               ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
001410  000771                        BR      MA1ONC
```

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

001412  005004          INITUD: CLR     R4                      ;R4 IS USED IF AN ERROR IS DETECTED
001414  012702  000001          MOV     #1,R2                   ;R2 = STEP INDICATOR REG FOR MSG'S
001420  005077  176362          CLR     @ADDR                   ;WRITE TO IP REGISTER TO INIT CONTROLLER
001424  012701  002260          MOV     #TIMER,R1               ;SET TIME OUT LIMIT
001430  017700  176616  1$:     MOV     @SAREG,R0               ;R0 HAS SA REGISTER DATA
001434  032700  100000          BIT     #<SA.ERR>,R0            ;CHECK FOR ERROR
001440  001007                  BNE     2$                      ;IF FOUND, GET OUT OF LOOP
001442  104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001446  104407  000000'                  BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001452  005301                  DEC     R1                      ;TIME OUT?
001454  001365                  BNE     1$                      ;IF NOT, LOOP
001456  000404                  BR      4$                      ;IF DONE, CONTINUE
001460  012703  004000  2$:     MOV     #SA.S1,R3               ;R3 = STEP 1 BIT
001464  000167  001150          JMP     ERROR1                  ;IF HERE, ERROR
001470  042700  173377  4$:     BIC     #^C<SA.S1!SA.DIA>,R0    ; CLEAR KDA50-Q DEPENDENT BITS
001474  022700  004400          CMP     #<SA.S1!SA.DIA>,R0      ;DID DATA COMPARE PROPERLY?
001500  001402                  BEQ     5$                      ;IF SO, CONTINUE
001502  000167  001126          JMP     ERROR3                  ;REPORT ERROR
                                STEP 2
001506  016705  176276  5$:     MOV     VECTOR,R5               ;VECTOR GIVEN
001512  006205                  ASR     R5                      ;SET TO APPROPRIATE VALUE
001514  006205                  ASR     R5                      ; = VECTOR/4
001516  052705  100200          BIS     #<SA.INT!BIT15>,R5      ;ACTIVATE INTERRUPTS & SET MSB FOR STEP 1
                                                                ;LEN'S ARE 0
001522  010500                  MOV     R5,R0                   ;STORE R5 IN R0 FOR SUBROUTINE
001524  012703  004000          MOV     #SA.S1,R3               ;R3 HAS STEP BIT FOR SUBROUTINE
001530  004767  001002          JSR     PC,SNDSTP               ;SEND STEP DATA
001534  042705  100000          BIC     #BIT15,R5               ;CLEAR MSB FOR COMPARE DATA
001540  042700  000200          BIC     #BIT07,R0               ;WAS BIT07 ONLY BIT SET?, SHOULD BE
001544  001404                  BEQ     6$
001546  052700  010200          BIS     #<SA.S2!BIT07>,R0       ;SET R0 TO REPORT THE ERROR
001552  000167  001056          JMP     ERROR3                  ;REPORT ERROR
001556  016700  176514  6$:     MOV     RSPPP,R0                ;R0 GETS PHYSICAL ADDRESS
001562  004767  000750          JSR     PC,SNDSTP               ;SEND STEP DATA
```

MODULE CODE

```
                                                        BIC     #177400,R5              ;HIGH BYTE CLEARED
                                                        CMP     R5,R0                   ;CHECK ECHO DATA
                                                        BEQ     7$                      ;IF OK, SKIP
                                                        JMP     ERROR3                  ;IF NOT, REPORT ERROR
                                                7$:
                                                    STEP 3
                                                        MOV     RSPEP,R0                ; SEND THE EXTENDED ADDRESS BITS
                                                        JSR     PC,SNDSTP               ; SEND STEP DATA
                                                        MOV     @RSPONC-4,R0            ; R0 -> RING ENVELOP
                                                    STEP 4
                                                8$:     TST     (R0)+                   ;IS THE RING ENTRY = 0?
                                                        BEQ     9$                      ;
                                                        JMP     ERROR5                  ;IF NOT, ERROR
                                                9$:     CMP     @CMDREF,R0              ;IS R0 POINT PAST THE RINGS?
                                                        BNE     8$                      ;IF NOT, LOOP
                                                        MOV     SR2,R0                  ;R0 = BURST VALUE
                                                        CLC                             ;CLEAR CARRY
                                                        ASL     R0                      ;ALIGN BURST FOR STEP 4
                                                        ASL     R0                      ;  "
                                                        BIS     @SA.GO,R0               ;SET GO BET
                                                        MOV     R0,@SAREG               ;SEND DATA TO CONTROLLER/INIT DONE
                                                        MOV     CPAKPP,COMMND           ;STORE ADDRESS IN THE RING
                                                        MOV     CPAKEP,COMMND+2         ;MOVE ADJUSTED EA INTO RING

                                                        MOV     RPAKPP,RSPONC           ;STORE ADDRESS IN THE RING
                                                        MOV     RPAKEP,RSPONC+2         ;MOVE ADJUSTED EA INTO RING
                                                        MOV     @NTRUPT,@VECTOR         ;STORE INTERRUPT ADDRESS IN VECTOR
                                                        CLR     TRY                     ;CLEAR TRY SO DRIVE WILL
                                                                                        ;GO BACK ONLINE IF NECESSARY
                                                        RTS     PC

                                                ;++
                                                ;       GETWB - GET WRITE BUFFER
                                                ;--
                                                GETWB:
                                                        GWBUF$, BEGIN                   ;GET WRITE BUFFER INFORMATION
                                                        BIT     @QMON22,RES2            ; IF NOT USING Q-22 MONITOR.
                                                        BEQ     11$                     ;   USE 18 BIT ADDRESSING
                                                        BIT     @ADDR22,CONFIG          ; IF 22-BIT QBUS ADDRESSING,
                                                        BNE     12$                     ;   CALCULATE PHYSICAL ADDRESS
                                                11$:    MOV     WBUFPA,WBUFPP           ; CONVERT FROM 18 BIT
                                                        MOV     WBUFEA,R0               ;   PSEUDO ADDRESS
                                                        JSR     PC,ASR04                ;     TO 18 BIT
                                                        MOV     R0,WBUFEP               ;       PHYSICAL ADDRESS
                                                        BR      20$                     ;
                                                12$:    MOV     WBUFPA,PA               ; SET UP FOR
                                                        MOV     WBUFEA,EA               ;   MAP22 CALL
                                                        MAP22$, BEGIN,PA                ; GET 22-BIT ADDR FROM 18-BIT ADDR
                                                        MOV     PA22,WBUFPP             ;   PHYSICAL ADDRESS
                                                        MOV     EA22,WBUFEP             ;
                                                20$:    RTS     PC                      ; RETURN FROM SUBROUTINE

                                                ;++
                                                ;       CVTADR - CONVERT 16 BIT ADDRESS TO 18 OR 22 BIT ADDRESS
                                                ;--
                                                CVTADR: MOV     @RSPONC,VA              ; CONVERT RESPONCE RING ADDRESS
```

MODULE CODE

```
                                        GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
                                        MOV     EA,RSPEA                ; SAVE EA BITS
                                        MOV     PA,RSPPA                ; SAVE PA BITS
                                        BIT     #QMON22,RES2            ; IF NOT USING Q-22 MONITOR,
                                        BEQ     11$                     ;   USE 18 BIT ADDRESSING
                                        BIT     #ADDR22,CONFIG          ; IF 22-BIT QBUS ADDRESSING,
                                        BNE     12$                     ;   CALCULATE PHYSICAL ADDRESS
002040  104415  000000' 000472'        11$:    MOV     PA,RSPPP        ; CONVERT FROM 18 BIT
002046  016767  176424  176220                  MOV     EA,R0           ;   PSEUDO ADDRESS
002054  016767  176414  176210                  JSR     PC,ASR04        ;     TO 18 BIT
002062  032767  000010  175770                  MOV     R0,RSPEP        ;       PHYSICAL ADDRESS
002070  001404                                  BR      20$             ;
002072  032767  001000  175756
002100  001012
002102  016767  176366  176166        12$:
002110  016700  176362                          MAP22$, BEGIN,PA        ; GET 22-BIT ADDR FROM 18-BIT ADDR
002114  004767  000404                          MOV     PA22,RSPPP      ;   PHYSICAL ADDRESS
002120  010067  176154                          MOV     EA22,RSPEP      ;
002124  000411
002126
        002126  104416  000000' 000474'
002134  016767  176340  176134
002142  016767  176334  176130
                                        20$:    MOV     RBUFVA,VA       ; CONVERT READ BUFFER ADDRESS
002150  016767  175750  176314                  GETPA$,BEGIN, VA        ;GET PHYSICAL ADDRESS FROM 16-BIT VA
002156  104415  000000' 000472'                 MOV     EA,RBUFEA       ; SAVE EA BITS
002164  016767  176306  175736                  MOV     PA,RBUFPA       ; SAVE PA BITS
002172  016767  176276  175726                  BIT     #QMON22,RES2    ; IF NOT USING Q-22 MONITOR,
002200  032767  000010  175652                  BEQ     21$             ;   USE 18 BIT ADDRESSING
002206  001404                                  BIT     #ADDR22,CONFIG  ; IF 22-BIT QBUS ADDRESSING,
002210  032767  001000  175640                  BNE     22$             ;   CALCULATE PHYSICAL ADDRESS
002216  001012                                  MOV     PA,RBUFPP       ; CONVERT FROM 18 BIT
002220  016767  176250  176256        21$:    MOV     EA,R0           ;   PSEUDO ADDRESS
002226  016700  176244                          JSR     PC,ASR04        ;     TO 18 BIT
002232  004767  000266                          MOV     R0,RBUFEP       ;       PHYSICAL ADDRESS
002236  010067  176244                          BR      30$             ;
002242  000411
002244                                  22$:
        002244  104416  000000' 000474'       MAP22$, BEGIN,PA        ; GET 22-BIT ADDR FROM 18-BIT ADDR
002252  016767  176222  176224                  MOV     PA22,RBUFPP     ;   PHYSICAL ADDRESS
002260  016767  176216  176220                  MOV     EA22,RBUFEP     ;
                                        30$:    MOV     #CMPACK,VA      ; CONVERT COMMAND PACKET ADDRESS
002266  012767  000402' 176176                  GETPA$,BEGIN, VA        ;GET PHYSICAL ADDRESS FROM 16-BIT VA
002274  104415  000000' 000472'                 MOV     EA,CPAKEA       ; SAVE EA BITS
002302  016767  176170  176154                  MOV     PA,CPAKPA       ; SAVE PA BITS
002310  016767  176160  176144                  BIT     #QMON22,RES2    ; IF NOT USING Q-22 MONITOR,
002316  032767  000010  175534                  BEQ     31$             ;   USE 18 BIT ADDRESSING
002324  001404                                  BIT     #ADDR22,CONFIG  ; IF 22-BIT QBUS ADDRESSING,
002326  032767  001000  175522                  BNE     32$             ;   CALCULATE PHYSICAL ADDRESS
002334  001012                                  MOV     PA,CPAKPP       ; CONVERT FROM 18 BIT
002336  016767  176132  176122        31$:    MOV     EA,R0           ;   PSEUDO ADDRESS
002344  016700  176126                          JSR     PC,ASR04        ;     TO 18 BIT
002350  004767  000150                          MOV     R0,CPAKEP       ;       PHYSICAL ADDRESS
002354  010067  176110                          BR      40$             ;
002360  000411
002362                                  32$:
        002362  104416  000000' 000474'       MAP22$, BEGIN,PA        ; GET 22-BIT ADDR FROM 18-BIT ADDR
002370  016767  176104  176070                  MOV     PA22,CPAKPP     ;   PHYSICAL ADDRESS
002376  016767  176100  176064                  MOV     EA22,CPAKEP     ;
                                        40$:    MOV     #RSPACK,VA      ; CONVERT RESPONSE PACKET ADDRESS
002404  012767  000306' 176060                  GETPA$,BEGIN, VA        ;GET PHYSICAL ADDRESS FROM 16-BIT VA
002412  104415  000000' 000472'
```

```
    781 002420  016767  176052  175742          MOV     EA,RPAKEA               ; SAVE EA BITS
    782 002426  016767  176042  175732          MOV     PA,RPAKPA               ; SAVE PA BITS
    783 002434  032767  000010  175416          BIT     #QMON22,RES2            ; IF NOT USING Q-22 MONITOR,
    784 002442  001404                          BEQ     41$                     ;   USE 18 BIT ADDRESSING
    785 002444  032767  001000  175404          BIT     #ADDR22,CONFIG          ; IF 22-BIT QBUS ADDRESSING,
    786 002452  001012                          BNE     42$                     ;   CALCULATE PHYSICAL ADDRESS
    787 002454  016767  176014  175710  41$:    MOV     PA,RPAKPP               ; CONVERT FROM 18 BIT
    788 002462  016700  176010                  MOV     EA,R0                   ;   PSEUDO ADDRESS
    789 002466  004767  000032                  JSR     PC,ASR04                ;     TO 18 BIT
    790 002472  010067  175676                  MOV     R0,RPAKEP               ;       PHYSICAL ADDRESS
    791 002476  000411                          BR      50$                     ;
    792 002500                          42$:
        002500  104416  000000'  000474'        MAP22$, BEGIN,PA                ; GET 22-BIT ADDR FROM 18-BIT ADDR
    793 002506  016767  175766  175656          MOV     PA22,RPAKPP             ;   PHYSICAL ADDRESS
    794 002514  016767  175762  175652          MOV     EA22,RPAKEP             ;
    795
    796 002522  000207                  50$:    RTS     PC                      ; RETURN FROM SUBROUTINE
    797
    798                         ;*************************************************************************
    799                         ;                                                                        ;
    800                         ;       ASR04                                                            ;
    801                         ;               ARITHMETIC SHIFT RIGHT REG 0 FOUR TIMES                  ;
    802                         ;                                                                        ;
    803                         ;       EXTENDED ADDRESS BITS (16 & 17) ARE SET IN BIT POSITION 4 & 5    ;
    804                         ;       RESPECTIVELY.  SHIFT RIGHT FOUR TIMES TO REPOSTION THE VALUE     ;
    805                         ;                                                                        ;
    806                         ;       INPUT   R0 = UNADJUSTED EXTENDED ADDRESS BITS                    ;
    807                         ;                                                                        ;
    808                         ;       OUTPUT  R0 = ADJUSTED EXTENDED ADDRESS BITS                      ;
    809                         ;                                                                        ;
    810                         ;*************************************************************************
    811 002524                  ASR04:
    812 002524  006200                  ASR     R0                      ;SHIFT 10
    813 002526  006200                  ASR     R0                      ; SHIFT 4
    814 002530  006200                  ASR     R0                      ;  SHIFT 2
    815 002532  006200                  ASR     R0                      ;   SHIFT 1
    816 002534  000207                  RTS     PC                      ;RETURN
    817
    818                         ;*************************************************************************
    819                         ;
    820                         ;       SEND STEP DATA
    821                         ;
    822                         ;       INPUT:  R0 HAS DATA TO BE SENT TO CONTROLLER FOR STEP
    823                         ;               R3 HAS PREVIOUS STEP FLAG SET
    824                         ;
    825                         ;       OUTPUT: R0 HAS DATA SENT FROM CONTROLLER TO HOST FOR ECHO AND NEXT STEP
    826                         ;               R3 HAS CURRENT STEP FLAG SET
    827                         ;
    828                         ;*************************************************************************
    829 002536  016701  175246  SNDSTP: MOV     VECTOR,R1               ;
    830 002542  012721  002562'         MOV     #INTA,(R1)+             ;SET UP INTERRUPT HANDLER ADDRESS
    831 002546  116711  175240          MOVB    BR1,(R1)                ;SET PRIORTITY LEVEL
    832 002552  010077  175474          MOV     R0,@SAREG               ;SEND STEP1 WRITE FORMMATED DATA
    833
    834 002556  104400  000000'         EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
                                        INTA:
                                        ;-------------------------------------------------------------------
002562  000004  000000' 002570'         PIRQ$,BEGIN,3$          ; QUEUE UP TO CONTINUE AT 3$ AND RTI
                                        ;-------------------------------------------------------------------
                                        3$:
002570                                  MOV     @SAREG,R0               ;GET STEP N FORMATTED DATA
002570  017700  175456                  BIT     #SA.ERR,R0              ;TEST FOR ERROR
002574  032700  100000                  BNE     ERROR1                  ;IF NOT OK, REPORT
002600  001017                          INC     R2                      ;SET STEP REGISTER
002602  005202                          ASL     R3                      ;R3 HAS STEP BIT PROPERLY SET
002604  006303                          BIT     R3,R0                   ;WAS STEP N SET?
002606  030300                          BNE     4$                      ;IF SO, CONTINUE
002610  001002                          JMP     ERROR2                  ;IF NOT CORRECT STEP, ERROR
002612  000167  000020                  BIC     R3,R0           4$:     ;CLEAR THE STEP BIT, FOR COMPARE
002616  040300                          RTS     PC                      ;RETURN
002620  000207
```

```
                                ;*****************************************************************************
                                ;                                                                            *
                                ;       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                                                           *
                                ;                                                                            *
                                ;*****************************************************************************
002622                          ERROR5:
002622  104403  000000' 006062'         MSGN$,BEGIN,ZERO            ;ASCII MESSAGE CALL WITH COMMON HEADER
002630  104410  000000'                 END$,BEGIN                  ;
                                
002634  005204                  ERROR3: INC     R4                          ;R4 = 3 FOR ERROR3
002636  005204                  ERROR2: INC     R4                          ;R4 = 2 FOR ERROR2
002640  005204                  ERROR1: INC     R4                          ;R4 = 1 FOR ERROR1
002642  010267  175646                 MOV     R2,NUM                      ;STORE STEP REG IN A NUMBER FOR CONVRT
                                        ;*******************************************************************
                                                                            ;CONVERT NUM TO ASCII AND
                                                                            ;STORE AT ADR2
002646  104420  000000' 000514'         OTOA$,BEGIN,NUM,ADR2
002654  000535'
                                        ;*******************************************************************
002656  017767  175370  175630         MOV     @SAREG,NUM                  ;STORE VALUE IN A NUMBER
                                        ;*******************************************************************
                                                                            ;CONVERT NUM TO ASCII AND
                                                                            ;STORE AT ADR1
002664  104420  000000' 000514'         OTOA$,BEGIN,NUM,ADR1
002672  000526'
                                        ;*******************************************************************
                                
                                        DEC     R4                          ;ERROR 1?
002674  005304                          BNE     1$                          ;IF NOT, CHECK IF IT IS THE NEXT ERROR
002676  001003
002700  104403  000000' 005672'         MSGN$,BEGIN,INITE1          ;ASCII MESSAGE CALL WITH COMMON HEADER
002706                          1$:
                                        DEC     R4                          ;ERROR 2?
002706  005304                          BNE     2$                          ;IF NOT, CHECK IF IT IS THE NEXT ERROR
002710  001003
002712  104403  000000' 005716'         MSGN$,BEGIN,INITE2          ;ASCII MESSAGE CALL WITH COMMON HEADER
002720                          2$:
002720  005304                  DEC     R4                          ;ERROR 3?
```

```
                                        BNE     3$                      ;IF NOT, CHECK IF IT IS THE NEXT ERROR
894 002722  001003
895 002724  104403  000000' 005724'     MSGN$,BEGIN,INITE3      ;ASCII MESSAGE CALL WITH COMMON HEADER
896 002732                          3$:
897                                     ;*************************************************************************
                                        ;CONVERT ADDR TO ASCII AND
                                        ;STORE AT ADR3
    002732  104420  000000' 000006'     OTOA$,BEGIN,ADDR,ADR3
    002740  000544'
                                        ;*************************************************************************
                                        ;*************************************************************************
898
    002742  104405  000000' 000000     HRDER$,BEGIN,NULL       ;
                                        ;*************************************************************************
899 002750  104403  000000' 005700'     MSGN$,BEGIN,INITER      ;ASCII MESSAGE CALL WITH COMMON HEADER
900 002756  005267  175652              INC     TRY                     ;INCREMENT RETRY COUNT
901 002762  022767  000004  175644      CMP     #RLIM,TRY               ;IS THE RETRY COUNT EXCEEDED?
902 002770  001402                      BEQ     6$                      ;IF SO, END IT
903 002772  000167  175712              JMP     START                   ;IF NOT, TRY AGAIN
904 002776                          6$:
    002776  104403  000000' 006056'     MSGN$,BEGIN,ABORT       ;ASCII MESSAGE CALL WITH COMMON HEADER
905 003004  104410  000000'             END$,BEGIN              ;
906
```

```
                                        ;*****************************************************************************
                                        ;
                                        ;       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.
                                        ;
                                        ;*****************************************************************************
003010                                  SETUP:
                                        ; ***   SET CONTRL CHAR AND WAIT FOR THE ATTENTION MESSAGES
003010  004767  001650                          JSR     PC,SCC                  ;SET CONTROLLER CHARACTERISTICS
003014  005367  175250                          DEC     CMDREF                  ;ONLY SET UP AT BEGINNING OF MODULE
003020  001110                                  BNE     19$                     ; (USE DRIVES FOUND AT BEGINNING)
003022  012703  000001                          MOV     #1,R3                   ;INITIAL PORTID VALUE
003026  012704  000644'                        MOV     #TABLEW,R4              ;R4 -> TABLEW
003032  011467  175574                          MOV     (R4),UNITNO             ;INITIAL UNITNO IN TABLEW
003036  016714  175570                  1$:     MOV     UNITNO,(R4)             ;UNIT NO SET IN TABLEW;READY TO TEST
003042  010367  175570                          MOV     R3,PORTID               ;PORT ID SET
003046  010364  000002                          MOV     R3,2(R4)                ;PORTID SET IN TABLEW
003052  012764  177777  000004                  MOV     #177777,4(R4)           ;INSERT NEW -1,-1 FOR LAST ENTRY
003060  016464  000004  000006                  MOV     4(R4),6(R4)             ;  OF THE TABLEW
003066  012767  002400  175546                  MOV     #2400,WORK              ;WORK = RETRY LIMIT
003074  004767  001526                  3$:     JSR     PC,GTSTAT               ;GET STATUS, GET NEXT UNIT NUMBER
003100  103006                                  BCC     7$                      ;OK, CONTINUE
003102  005367  175534                          DEC     WORK                    ;ELSE IF OFFLINE, DECR COUNT
003106  001372                                  BNE     3$                      ;IF COUNT > 0, TRY AGAIN.
003110  004767  000774                  5$:     JSR     PC,DROP2                ;DROP THE DRIVE
003114  000437                                  BR      17$                     ;TRY NEXT UNIT
003116  016767  175170  175506          7$:     MOV     P.UNIT+RSPACK,UNITNO    ;UNIT NUMBER FROM RESPONCE PACKET IN UNITNO
                                        ; *** CHECK FOR CASE WHERE THE MORE UNITS THEN DRIVES HAVE BEEN SPECIFIED.
                                        ; *** NEXT UNIT MODIFIER WILL GIVE A DUPLICATE UNIT NUMBER.
003124  012702  000644'                          MOV     #TABLEW,R2              ;R2 -> TABLE TO FIND DUPLICATE
003130  012705  000001                          MOV     #1,R5                   ;R5 IS TEMP PORTID
003134  020227  000704'                  9$:     CMP     R2,#TEND                ;REACHED THE BOTTOM?
003140  001420                                  BEQ     15$                     ;IF SO, EXIT
003142  020305                                  CMP     R3,R5                   ;REACHED THE LATEST ENTRY?
003144  001416                                  BEQ     15$                     ;IF SO, EXIT
003146  026712  175460                          CMP     UNITNO,(R2)             ;DO WE HAVE A DUPLICATE UNIT NUMBER?
003152  001404                                  BEQ     13$                     ;IF SO, ERROR
003154  062702  000004                  11$:    ADD     #4,R2                   ;IF NOT, POINT TO NEXT POINTER
003160  006305                                  ASL     R5
003162  000764                                  BR      9$                      ;AND CONTINUE
003164  011467  175442                  13$:    MOV     (R4),UNITNO             ;DROP DRIVE FROM TABLE
003170  010367  175442                          MOV     R3,PORTID               ;
003174  004767  000720                          JSR     PC,DROP3                ;AND DROP IT
003200  000405                                  BR      17$
003202                                  15$:
                                        ; ***
```

MODULE CODE

```
                                        CMP     UNITNO,(R4)             ;IS THE UNITNO CORRECT?
                                        BEQ     17$                     ;IF SO, CHECK FOR NEXT UNIT
                                        MOV     UNITNO,(R4)             ;ELSE, CORRECT THE UNIT NUMBER IN TABLE
                                17$:
                                        ASL     R3                      ;NEXT PORTID SET
                                        CMP     DVICE,R3                ;DONE?
                                        BMI     19$                     ;IF R3 > DVICE, ALL DESIRED DRIVES ARE FOUND.
                                        INC     UNITNO                  ;NEXT UNITNO SET
                                        ADD     #4,R4                   ;POINT TO NEXT ENTRY TO TEST DRIVE
                                        CMP     #TEND,R4                ;POINT TO END? IF SO, TABLE FULL
                                        BHI     1$                      ;IF R4 NOT REACHED END, GO TEST
                                19$:
                                        RTS     PC

                                ;*******************************************************************************
                                ;
                                ;       TSTOFL
                                ;               TEST TO SEE WHAT KIND OF AN OFFLINE CONDITION HAS OCCURED.
                                ;
                                ;*******************************************************************************
                                TSTOFL: CMP     #ST.OFL,R0              ;WAS THE DRIVE FOUND OFFLINE?
                                        BEQ     10$                     ;CHECK WHAT KIND OF OFFLINE
                                        CMP     #ST.DRV,R0              ;WAS IT A DRIVE ERROR? -> SDI?
                                        BNE     13$                     ;IF IT WAS NOT, ERROR (DROP DRIVE)
                                10$:    BIT     #<SC.NVL+SC.DIS+SC.DUP+SC.IOP>,P.STS+RSPACK     ;WERE ANY OF THESE BITS SET?
                                                                        ; = NO VOLUME MOUNTED, UNIT DIABLED BY FIELD SREVICE
                                                                        ;   OR DUPLICATE UNIT NUMBER OR UNIT INOPERATIVE
                                        BNE     12$                     ;IF SO, EXIT
                                        BIT     #^C<SC.NVL+SC.DIS+SC.DUP+SC.IOP+ST.MSK>,P.STS+RSPACK ; ANY OTHER DATA?
                                        BNE     13$                     ;IF SO, DROP
                                12$:    CLC                             ;CLEAR CARRY
                                        RTS     PC                      ;RETURN
                                13$:    SEC                             ;SET CARRY, DRIVE WAS FOUND TO BE OFFLINE
                                                                        ; OR ANOTHER ERROR
                                        JSR     PC,ERRORH               ;REPORT ERROR
                                        RTS     PC                      ;RETURN
```

MODULE CODE

```
1000                                    ;*****************************************************************************
1001                                    ;                                                                            *
1002                                    ;       CYCLE DISK                                                           *
1003                                    ;                                                                            *
1004                                    ;       DO THE DISK CYCLE                                                    *
1005                                    ;               DO GET STATUS COMMANDS TO ASSURE THAT THE DRIVE              *
1006                                    ;                       IS AVAILABLE (FOR DUAL PORTING)                      *
1007                                    ;               CHECK DRIVE TO BE ONLINE                                     *
1008                                    ;               IF TRUE                                                      *
1009                                    ;                    PICK THE BLOCK                                          *
1010                                    ;                    WRITE                                                   *
1011                                    ;                    READ                                                    *
1012                                    ;                    DATA CHECK                                              *
1013                                    ;                    MAKE THE DRIVE AVAILABLE                                *
1014                                    ;               ELSE DROP DRIVE                                              *
1015                                    ;                                                                            *
1016                                    ;*****************************************************************************
1017 003314                            CYCLED:
1018 003314  C32767  001000  174474             BIT     @SR.DUA,SR1             ;DUAL PORT?
1019 003322  001004                             BNE     2$                      ;IF NOT, CONTINUE
1020                                    ; *** CHECK IF WE DO ONLINE FOR THE FIRST TIME.
1021 003324  005767  175304                     TST     TRY                     ;IF TRY HAS SET MSB, DON'T DO ONLINE
1022 003330  100443                             BMI     16$                     ; DON'T DO ONLINE
1023 003332  000422                             BR      10$                     ; ELSE DO ONLINE (1ST TIME THROUGH IN THIS PASS)
1024                                    ; ***
1025                                    ; *** DO GET STATUS COMMANDS TO ASSURE THE DRIVE IS AVAILABLE TO THE CONTROLLER
1026                                    ; ***  FOR DUAL PORTING.
1027                                    ; ***
1028 003334  012701  000010             2$:     MOV     @10,R1                  ;R1 = # OF GET STATUS TO DO
1029 003340  004767  001262             4$:     JSR     PC,GTSTAT               ;IS THE DRIVE OFFLINE?
1030 003344  103013                             BCC     6$                      ;IF ALL OK, DO THE CYCLE
1031 003346  004767  177672                     JSR     PC,TSTOFL               ;ELSE, CHECK IF OFFLINE
1032 003352  103507                             BCS     24$                     ;IF IT ERRED, DROP THE DRIVE
1033                                    ; *** HANDLE OFF LINE DRIVE, WAIT FOR AVAILABLE ATTENTION MESSAGE
1034 003354  005067  175144                     CLR     EXPAV                   ;EXPECT AN AVAILABLE ATTENTION MESSAGE
1035 003360  052767  140000  174674             BIS     @<RG.OWN+RG.FLG>,RSPONC+2       ;SET RING FOR ATTN MESSAGE
1036 003366  004767  001426                     JSR     PC,INTERP               ;WAIT FOR MESSAGE
1037                                                                            ; 2ND ATTENTION MESSAGE
1038 003372  000402                             BR      10$
1039 003374  005301                     6$:     DEC     R1                      ;DONE?
1040 003376  001360                             BNE     4$                      ;IF NOT DONE, TRY AGAIN
1041 003400  004767  001316             10$:    JSR     PC,ONLINE               ;DO AND ONLINE COMMAND
1042 003404  103753                             BCS     2$                      ;IF CARRY WAS SET, TRY AGAIN
1043 003406  016767  174742  175210     14$:    MOV     P.UNSZ+2+RSPACK,UNSZH   ;IS THE UNIT SIZE HI ADDRESS
1044 003414  016767  174732  175200             MOV     P.UNSZ+RSPACK,UNSZL     ;GET UNIT SIZE/IS IT = 0?
1045 003422  001006                             BNE     16$                     ;IF NOT ZERO, CONTINUE WITH ITERATION
1046 003424  005767  175174                     TST     UNSZH                   ;IS UNSZH ALSO 0?
1047 003430  001731                             BEQ     CYCLED                  ;IF 0, TRY TO BRING ONLINE AGAIN
1048                                    ; *** SET MSB OF TRY TO SHOW THAT INITIAL ONLINE IS DONE
1049 003432  012767  100000  175174             MOV     @100000,TRY
1050                                    ;*****************************************************************************
1051                                    ;                                                                            *
1052                                    ;       THE FOLLOWING SEQMENT SETS THE LIMIT FOR THE UNIT SIZE.              *
1053                                    ;       THE VALUE (UNIT SIZE - (WRITE BUFFER SIZE/NORMAL BLOCK SIZE))        *
1054                                    ;
```

MODULE CODE

```
                                        ;       IS THE LAST SECTOR POSSIBLE TO RIGHT TO.                        ;
                                        ;                                                                       ;
                                        ;***********************************************************************
                                        16$:
003440  016700  174476                          MOV     WBUFSZ,R0               ;WBUFSZ IN R0 AS A LIMIT
003440  005001                                  CLR     R1                      ;R1 = # OF BLOCKS
003444  005201                          18$:    INC     R1                      ;INCREMENT THE # OF BLOCKS
003446  162700  000400                          SUB     #400,R0                 ;DECREMENT A BLOCK
003450  100374                                  BPL     18$                     ;BR IF > 0
003454  160167  175140                          SUB     R1,UNSZL                ;ADJUST THE UNIT SIZE
                                        ; *** NOW PICK WHICH BLOCK TO WRITE TO
003456  004767  000156                          JSR     PC,PICKBK               ;ELSE SELECT A SECTOR TO TEST
003462  004767  001134                          JSR     PC,GTSTAT               ;DID WE NOT GET THE DRIVE ONLINE?
003466  103720                                  BCS     2$                      ;IF WE DID NOT, GO BACK TO TOP AND TRY AGAIN
003472  022700  000004                          CMP     #ST.AVL,R0              ;IS IT AVAILABLE?
003474  001715                                  BEQ     2$                      ;IF SO, GO BACK TO TOP AND TRY AGAIN
                                        ; *** WRITE TO THE BLOCK SELECTED
003500  004767  000720                          JSR     PC,WRITE                ;WRITE THE DATA FOR USER DEFINED # OF WORDS
003502  103007                                  BCC     19$                     ;IF OK, CONTINUE
003506  032767  001000  174300                  BIT     #SR.DUA,SR1             ;ARE WE DOING DUAL PORT?
003510  001306                                  BNE     2$                      ;IF YES, RETRY
003516  004767  001712                          JSR     PC,ERRORH               ;ELSE, HARD ERROR
003520  000421                                  BR      22$                     ;AND EXIT; BCS 22$ ;IF ERROR, EXIT
                                        ; *** READ IT BACK
003524  004767  000730                  19$:    JSR     PC,READ                 ;READ A BLOCK
003526  103416                                  BCS     22$                     ;IF ERROR, EXIT
003532  032767  004000  174254                  BIT     #SR.CMP,SR1             ;DO A DATA COMPARE?
003534  001004                                  BNE     20$                     ;IF NOT, SKIP THE COMPARE
                                        ; *** COMPARE DATA
003542  104412  000000' 000126'                CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
003544  003552  003554'                         .+2                     ; IF ERROR, CONTINUE
003554  032767  001000  174234          20$:    BIT     #SR.DUA,SR1             ;DO WE DO AN AVAILABLE?
003562  001402                                  BEQ     22$                     ;IF NOT(BIT NOT SET) SKIP AVAILABLE
                                        ; *** MAKE THE DRIVE AVAILABLE
003564  004767  001014                          JSR     PC,AVAILB               ;RELEASE THE DRIVE
003570  000241                          22$:    CLC                             ;EVERY THING WAS OK
                                                                                ;WASTE A LITTLE TIME SO OTHER
                                                                                ;  CONTROLLER CAN GRAB DRIVE
003572  000207                          24$:    RTS     PC                      ;RETURN

                                        ; *** SUBROUTINE TO WAIT FOR AN INTERRUPT
                                        ; *** RETURNS AFTER THE INTERRUPT OCCURS
                                        DOINTR:
003574                                          CLR     EXPAV                   ;EXPECT AN AVAILABLE ATTENTION MESSAGE
003574  005067  174724                          BIS     #<RG.OWN+RG.FLG>,RSPONC+2       ; SET OWN AND FLAG FOR RESPONCE RING
003600  052767  140000  174454                  JMP     INTERP                  ;WAIT FOR ATTENTION MESSAGE & RETURN
003606  000167  001206

                                        ;***********************************************************************
                                        ;                                                                       ;
                                        ;       DISKLESS CYCLE                                                  ;
                                        ;               DO A MAITENENCE WRITE                                   ;
                                        ;               AND A MAITENENCE READ                                   ;
                                        ;               AND CHECK THE DATA                                      ;
                                        ;                                                                       ;
                                        ;***********************************************************************
```

MODULE CODE

```
1109 003612                                CYCLEL:
1110 003612  004767  000470                       JSR     PC,MAITW                ;DO A MAINTENENCE WRITE
1111 003616  004767  000430                       JSR     PC,MAITR                ;DO A MAINTENENCE READ
1112 003622  032767  004000  174166               BIT     @SR.CMP,SR1             ;DO A DATA COMPARE?
1113 003630  001004                               BNE     21$                     ;IF NOT, SKIP THE COMPARE
1114 003632  104412  000000' 000126'              CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
     003640  003642'                               .+2                     ; IF ERROR, CONTINUE
1115 003642                                21$:
1116 003642  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)                       *
                                ;                                                                            *
                                ;*****************************************************************************
003644                  PICKBK: BIT     #SR.SEQ,SR1             ;CHECK SR1 FOR RANDOM ACCESS MODE
003644  032767  002000  174144          BEQ     SEQACC          ;BR IF SEQUENTIAL ACCESS
003652  001467
003654                  RANACC: RAND$,BEGIN
003654  104417  000000'                 MOV     RANNUM,-(SP)    ;GENERATE THE SECTOR ADDRESS
003660  016746  174170
003664  104417  000000'                 RAND$,BEGIN
003670  016746  174160                  MOV     RANNUM,-(SP)    ;GENERATE THE SECTOR ADDRESS
                                ;
                                ;       ADJUST HI ADDRESS FIRST
                                ;
003674  000241                          CLC                     ;CLEAR CARRY FOR ROTATE
003676  042716  100000                  BIC     #100000,(SP)    ;CLEAR UPPER BIT MAKES SURE VALUE'S +
003702  012667  174712                  MOV     (SP)+,SECH      ;STORE IN SECTOR HI ADDRESS
003706  005767  174712                  TST     UNSZH           ;IS THE MAX SIZE 0?
003712  001430                          BEQ     3$              ;IF 0, GET LOW SECTOR ADDRESS
                                ;  ***  UNSZH > 0 IF CODE FALLS THROUGH HERE
003714  016700  174704                  MOV     UNSZH,R0        ;R0 = MAX VALUE
003720  005100                          COM     R0              ;R0 COMPLEMENT, NOW FIND MS ZERO
003722  012701  100000                  MOV     #100000,R1      ;R1 IS INDEX INTO MAX VALUE
003726  030100                  1$:     BIT     R1,R0           ;HAVE 0 YET?
003730  001403                          BEQ     2$              ;IF 1ST 0 REACHED, CLEAR REST OF THE BITS
003732  000241                          CLC                     ;CLEAR CARRY FOR ROR
003734  006001                          ROR     R1              ;POINT TO NEXT BIT
003736  000773                          BR      1$              ;BRANCH TO TEST AGAIN
003740  040100                  2$:     BIC     R1,R0           ;CLEAR REST OF THE BITS
003742  000241                          CLC                     ;CLEAR CARRY FOR ROR
003744  006001                          ROR     R1              ;IF R1 ROTATES INTO CARRY, R1 = 0
003746  001374                          BNE     2$              ;IF R1 NOT 0, MORE BITS TO CLEAR
003750  040067  174644                  BIC     R0,SECH         ;CLEAR UPPER BITS OF HIGH SECTOR VALUE
003754  026767  174640  174642          CMP     SECH,UNSZH      ;IF THE HIGH SECTOR VALUE > MAX VALUE?
003762  002420                          BLT     7$              ;IF <, EXIT
003764  001405                          BEQ     4$              ;IF =, TEST LOW ORDER VALUE
003766  006267  174626                  ASR     SECH            ;SECH = SECH/2 - CAN'T BE > MAX NOW
003772  000414                          BR      7$              ;EXIT
                                ;
                                ;       GET LOW SECTOR ADDRESS
                                ;
003774  005067  174620          3$:     CLR     SECH            ;CLEAR HI SECTOR SIZE
004000  005767  174616          4$:     TST     UNSZL           ;IS THE HIGHEST POSSIBLE = 0?
004004  001406                          BEQ     6$              ;IF TRUE, DON'T DO LOOP
004006  166716  174610          5$:     SUB     UNSZL,(SP)      ;ELSE, SECL = SECL - UNSZL (ADJUST)
004012  103375                          BCC     5$              ;IF UNSZL > SECL, LOOP
004014  066716  174602                  ADD     UNSZL,(SP)      ;ELSE SUBTRACTED ONCE TOO OFTEN
004020  000401                          BR      7$              ; AND EXIT
```

MODULE CODE

```
1173 004022  005016                    6$:     CLR     (SP)                    ;CLEAR LO SECTOR ADDRESS (IF HIGHEST POSSIBLE = 0)
1174 004024  012667  174566            7$:     MOV     (SP)+,SECL              ;SAVE LO SECTOR ADDRESS
1175 004030  000207                            RTS     PC                      ; RETURN
1176
1177
1178                                   ;GENERATE DISK ADDRESS BY SEQUENTIAL ADDRESSING
1179
1180 004032                            SEQACC:
1181 004032  005267  174560                    INC     SECL                    ;INCREMENT THE SECTOR ADDRESS
1182 004036  001405                            BEQ     16$                     ;BR IF ZERO
1183 004040  026767  174552  174554            CMP     SECL,UNSZL              ;OVER LIMIT?
1184 004046  103413                            BLO     18$                     ;BR IF LOWER
1185 004050  000402                            BR      17$                     ;SKIP THE INCREMENT
1186 004052                            16$:
1187 004052  005267  174542                    INC     SECH                    ;INCREMENT SECTOR HIGH ADDRESS
1188 004056                            17$:
1189 004056  026767  174536  174540            CMP     SECH,UNSZH              ;OVER LIMIT?
1190 004064  103404                            BLO     18$                     ;BR IF LOWER
1191 004066  C05067  174524                    CLR     SECL                    ;RESET THE STARTING SECTOR ADDRESS
1192 004072  005067  174522                    CLR     SECH                    ;
1193
1194 004076                            18$:
1195 004076  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                                         *
                        ;                                                                            *
                        ;*****************************************************************************
004100                  DROP1:
004100  012767  000001  174406          MOV     #1,NUM
004106  000407          DROP2:          BR      DROP4
004110                  DROP3:
004110  012767  000002  174376          MOV     #2,NUM
004116  000403          DROP4:          BR      DROP4
004120                                  MOV     #3,NUM
004120  012767  000003  174366          BIT     PORTID,DVICE            ;HAS THE DRIVE BEEN DROPPED, DON'T DROP AGAIN
004126  036767  174504  174474          BEQ     10$                     ;
004134  001445                          CMP     #177777,PORTID          ;IF DRIVE HAS BEEN DROPPED, DON'T DROP AGAIN
004136  022767  177777  174472                                          ;       (WILL ZERO DVICE PREMATURE)
                                        BEQ     10$                     ;IF =, DRIVE HAS BEEN DROPPED -> EXIT ROUTINE
004144  001441
                                        BIC     PORTID,DVICE            ;DROP THE DRIVE
004146  046767  174464  174454          ;*************************************************************
                                                                        ;CONVERT UNITNO TO ASCII AND
                                                                        ;STORE AT ADR2
004154  104421  000000' 000632'         BTOD$,BEGIN,UNITNO,ADR2
004162  000535'
                                        ;*************************************************************
                                        CLRB    ADR2+5
004164  105067  174352                  ;*************************************************************
                                                                        ;CONVERT PORTID TO ASCII AND
                                                                        ;STORE AT ADR1
004170  104420  000000' 000636'         OTOA$,BEGIN,PORTID,ADR1
004176  000526'
                                        ;*************************************************************
                                        MOV     #177777,2(R4)           ;DESELECT DRIVE SO IT WON'T BE USED AGAIN.
004200  012764  177777  000002          DEC     NUM                     ;DROPPED FOR WHICH ERROR?
004206  005367  174302                  BNE     1$                      ;IF NOT FOR ERRORS, CONTINUE
004212  001004                          MSGN$,BEGIN,DRP1        ;ASCII MESSAGE CALL WITH COMMON HEADER
004214  104403  000000' 005732'         BR      10$
004222  000412                  1$:     DEC     NUM                     ;WAS UNIT NOT FOUND?(NON EXISTENT UNIT)
004224  005367  174264                  BNE     2$                      ;IF NOT, CONTINUE
004230  001004                          MSGN$,BEGIN,DRP2        ;ASCII MESSAGE CALL WITH COMMON HEADER
004232  104403  000000' 005750'         BR      10$
004240  000403
                                2$:
004242                          MSGN$,BEGIN,DRP3        ;ASCII MESSAGE CALL WITH COMMON HEADER
004242  104403  000000' 005766'                                 ; ACTUAL UNITS FOUND
```

MODULE CODE

```
   1241 004250  000207                    10$:    RTS     PC
   1242
   1243
```

```
                                ;*****************************************************************************
                                ;                                                                            *
                                ;       MAITENENCE READ                                                      *
                                ;                                                                            *
                                ;       SET UP A PACKET WITH:                                                *
                                ;               OPCODE & MODIFIER                                            *
                                ;               REGION ID & REGION OFFSET                                    *
                                ;               READ BUFFER DESCRIPTOR                                       *
                                ;               BYTE COUNT                                                   *
                                ;       THEN SEND THE PACKET                                                 *
                                ;                                                                            *
                                ;*****************************************************************************
004252  004767  001066          MAITR:  JSR     PC,CLRPAK               ;CLEAR THE PACKETS
004256  012767  000030  174126          MOV     #OP.MRD,P.OPCD+CMPACK   ;SET THE OPCODE
004264  016767  174216  174132          MOV     RBUFEP,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR
004272  016767  174206  174122          MOV     RBUFPP,P.ADPA+CMPACK    ;
004300  016700  173626                  MOV     RBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
004304  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                                                 *
                                ;                                                                            *
                                ;*****************************************************************************
004306  004767  001032          MAITW:  JSR     PC,CLRPAK               ;CLEAR THE PACKETS
004312  012767  000031  174072          MOV     #OP.MWR,P.OPCD+CMPACK   ;SET THE OPCODE
004320  016767  174166  174076          MOV     WBUFEP,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR
004326  016767  174156  174066          MOV     WBUFPP,P.ADPA+CMPACK    ;
004334  026767  173602  174264          CMP     WBUFSZ,LIMIT            ;IS THE BUFFER SIZE > LIMIT?
004342  100403                          BMI     1$                      ;IF NOT, WBUFSZ IS OK
004344  016700  174256                  MOV     LIMIT,R0                ;STORE THE BUFFER SIZE IN WORDS
004350  000402                          BR      MAITP                   ;AND SKIP
004352  016700  173564          1$:     MOV     WBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
004356  006300                  MAITP:  ASL     R0                      ;MAKE IT NUMBER OF BYTES
004360  010067  174032                  MOV     R0,P.BCNT+CMPACK        ;SET WRITE BUFFER SIZE
004364  012767  000020  173710          MOV     #16.,RSPLEN             ;SET RESPONCE PACKET LENGTH
004372  012767  000044  173776          MOV     #36.,CMPLEN             ;SET COMMAND PACKET LENGTH
004400  012767  000001  174030          MOV     #1,P.RGID+CMPACK        ;SET REGION ID = 1
004406  012767  177777  173764          MOV     #177777,CMPVIR          ;SET COMMAND VIRTUAL CIRCUIT (-1 FOR DM)
004414  012767  177777  173662          MOV     #177777,RSPVIR          ;SET COMMAND VIRTUAL CIRCUIT
004422  000167  000322                  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.                                                *
                                ;                                                                            *
                                ;*****************************************************************************
004426  004767  000712          WRITE:  JSR     PC,CLRPAK               ;CLEAR PACKETS
004432  012767  000042  173752          MOV     #OP.WR,P.OPCD+CMPACK    ;SET THE OPCODE
004440  016700  173476          WRITEA: MOV     WBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
004444  016767  174040  173750          MOV     WBUFPP,P.ADPA+CMPACK    ;SET THE BUFFER DESCRIPTOR(PA)
004452  016767  174034  173744          MOV     WBUFEP,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR(EA)
004460  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.                                                *
                                ;                                                                            *
                                ;*****************************************************************************
004462  004767  000656          READ:   JSR     PC,CLRPAK               ;CLEAR PACKETS
004466  012767  000041  173716          MOV     #OP.RD,P.OPCD+CMPACK    ;SET THE OPCODE
004474  016700  173432                  MOV     RBUFSZ,R0               ;STORE THE BUFFER SIZE IN WORDS
004500  016767  174002  173716          MOV     RBUFEP,P.ADEA+CMPACK    ;SET THE BUFFER DESCRIPTOR
004506  016767  173772  173706          MOV     RBUFPP,P.ADPA+CMPACK    ;
004514  012767  000040  173560  READA:  MOV     #32.,RSPLEN             ;SET RESPONCE PACKET LENGTH
004522  012767  000040  173646          MOV     #32.,CMPLEN             ;SET COMMAND PACKET LENGTH
004530  006300                          ASL     R0                      ;MAKE IT NUMBER OF BYTES
004532  010067  173660                  MOV     R0,P.BCNT+CMPACK        ;SET READ BUFFER SIZE
004536  016767  174054  173672          MOV     SECL,P.LBN+CMPACK       ;SET LOGICAL BLOCK NUMBER
004544  016767  174050  173666          MOV     SECH,P.LBN+2+CMPACK     ;
004552  000476                          BR      SEND                    ;SEND THE PACKET
```

```
1338                                    ;*****************************************************************************
1339                                    ;                                                                            *
1340                                    ;       DETERMINE ACCESS PATHS                                               *
1341                                    ;                                                                            *
1342                                    ;       SET UP CODE, GO SEND PACKET                                          *
1343                                    ;                                                                            *
1344                                    ;*****************************************************************************
1345 004554  004767  000564            DAP:    JSR     PC,CLRPAK               ;CLEAR PACKETS
1346 004560  012767  000013  173624            MOV     #OP.DAP,P.OPCD+CMPACK   ;SET OPCODE
1347 004566  012767  000074  173506            MOV     #60.,RSPLEN             ;SET LENGTHS
1348 004574  012767  000074  173574            MOV     #60.,CMPLEN             ;
1349 004602  000462                            BR      SEND                    ;SEND THE PACKET
1350
1351                                    ;*****************************************************************************
1352                                    ;                                                                            *
1353                                    ;       AVAILABLE PACKET                                                     *
1354                                    ;                                                                            *
1355                                    ;       SET OP CODE AND MODIFIERS THEN SEND THE PACKET                       *
1356                                    ;                                                                            *
1357                                    ;*****************************************************************************
1358 004604  004767  000534            AVAILB: JSR     PC,CLRPAK               ;CLEAR PACKETS
1359 004610  012767  000010  173574            MOV     #OP.AVL,P.OPCD+CMPACK   ;SET THE OPCODE
1360 004616  012767  000014  173456            MOV     #12.,RSPLEN             ;SET RESPONCE PACKET LENGTH
1361 004624  000413                            BR      GTSTAA                  ;SEND THE PACKET
1362
1363                                    ;*****************************************************************************
1364                                    ;                                                                            *
1365                                    ;       GET UNIT STATUS                                                      *
1366                                    ;                                                                            *
1367                                    ;       SET OPCODE AND MODIFIER (FOR THEN NEXT UNIT                          *
1368                                    ;               THEN SEND THE PACKET                                         *
1369                                    ;                                                                            *
1370                                    ;*****************************************************************************
1371 004626  004767  000512            GTSTAT: JSR     PC,CLRPAK               ;CLEAR PACKETS
1372 004632  012767  000003  173552            MOV     #OP.GUS,P.OPCD+CMPACK   ;SET THE OPCODE
1373 004640  012767  000001  173546            MOV     #MD.NXU,P.MOD+CMPACK    ;CLEAR MODIFIERS
1374 004646  012767  000060  173426            MOV     #48.,RSPLEN             ;SET RESPONCE PACKET LENGTH
1375 004654  012767  000014  173514    GTSTAA: MOV     #12.,CMPLEN             ;SET COMMAND PACKET LENGTH
1376 004662  000432                            BR      SEND                    ;SEND THE PACKET
```

```
                                ;*****************************************************************************
                                ;                                                                            :
                                ;       SET CONTROLLER CHARACTERISTICS
                                ;
                                ;       SET OP CODE AND CONTROLLER FLAG (ENABLE ATTENSION MSGS)
                                ;       CLEAR MSCP VERSION, HOST TIMEOUT, USE FRACTION,
                                ;            AND ALL OF QUAD WORD TIME AND DATE.
                                ;       THEN SEND PACKET
                                ;                                                                           :
                                ;*****************************************************************************
004664                          SCC:
004664  004767  000454                JSR     PC,CLRPAK               ;GO CLEAR THE COMMAND PACKET
004670  012767  000040  173500        MOV     #32.,CMPLEN             ;SET UP COMMAND PACKET LENGTH
004676  012767  000034  173376        MOV     #28.,RSPLEN             ;SET UP RESPONCE PACKET LENGTH
004704  012767  000004  173500        MOV     #OP.SCC*P.OPCD+CMPACK   ;SET THE OPCODE
004712  012767  000200  173500        MOV     #CF.AVL*P.CNTF+CMPACK   ;SET THE CONTROLLER FLAGS
                                                                      ; TO ENABLE ATTENTION MSGS
004720  000413                        BR      SEND                    ;SEND THE PACKET

                                ;*****************************************************************************
                                ;                                                                            :
                                ;       ONLINE                                                               :
                                ;                                                                            :
                                ;       SET OPCODE, MODIFIERS, UNIT ID, HOST ID                              :
                                ;       SHADOW UNIT, ERROR FLAGS                                             :
                                ;       THEN SEND PACKET                                                     :
                                ;                                                                            :
                                ;*****************************************************************************
004722  004767  000416        ONLINE: JSR     PC,CLRPAK               ;CLEAR PACKETS
004726  012767  000040  173346        MOV     #32.,RSPLEN             ;SET RESPONCE PACKET LENGTH
004734  012767  000044  173434        MOV     #36.,CMPLEN             ;SET COMMAND PACKET LENGTH
004742  012767  000011  173442        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                         *
                                ;                                                                            *
                                ;*****************************************************************************
004750  005267  173314    SEND:   INC     CMDREF                  ;NEW COMMAND REFERENCE NUMBER
004754  001775                    BEQ     SEND                    ;COMMAND REF # CANNOT = 0
004756  016767  173306  173416      MOV     CMDREF,P.CRF+CMPACK     ;SET COMMAND REF NUMBER
004764  016767  173642  173414      MOV     UNITNO,P.UNIT+CMPACK    ;SET UNIT NUMBER
004772  042767  040000  173266      BIC     #RG.FLG,COMMND+2        ;CLEAR FLAG
005000  052767  100000  173260      BIS     #RG.OWN,COMMND+2        ;SET OWN FOR COMMAND RING
005006  052767  140000  173246      BIS     #<RG.OWN+RG.FLG>,RSPONC+2  ;SET OWN AND FLAG FOR MESSAGE RING
005014  005777  172766              TST     @ADDR                   ;FORCE POLLING TO PACKET
005020                      INTERP:
005020  104400  000000'            EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;
005024                      NTRUPT:
                                        ;-------------------------------------------------------------
005024  000004  000000' 005032'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
                                ;
005032                      1$:
005032  005067  173220              CLR     RINTR                   ;CLEAR INTERRUPT FLAG
005036  022767  000100  173252      CMP     #OP.AVA,P.OPCD+RSPACK   ;WAS AN AVAILABLE ATTENTION RECIEVED?
005044  001524                      BEQ     15$                     ;IF IT WAS, EXIT
005046  022767  000102  173242      CMP     #OP.ACP,P.OPCD+RSPACK   ;WAS THE ACCESS PATH ATTENTION RECIEVED?
005054  001527                      BEQ     16$                     ;IF IT WAS, GO PROCESS
                                                                    ; ELSE CHECK SUCCESS
005056                      2$:
005056  016700  173236              MOV     P.STS+RSPACK,R0         ; SUCCESS?
005062  001513                      BEQ     14$                     ;IF YES, EXIT
005064  042700  177740              BIC     #177740,R0              ;CLEAR UPPER 11 BITS OF SUB-STATUS
005070  001510                      BEQ     14$                     ;IF SUCCESS = 0, EXIT OK
005072  005067  173010              CLR     ERRTYP                  ;IF GOT HERE, ERROR
005076  122700  000013              CMPB    #ST.DRV,R0              ; DRIVE ERROR?
005102  001015                      BNE     3$                      ;IF NOT NEXT TEST
005104  032767  001000  172704      BIT     #SR.DUA,SR1             ;ARE WE DUAL PORTING?
005112  001472                      BEQ     12$                     ;IF NOT, GO REPORT ERROR/ELSE EXPECTED
005114  022767  000053  173176      CMP     #<ST.DRV+SC.STO>,P.STS+RSPACK   ;IS IT AN SDI RESPONCE TIMEOUT?
005122  001464                      BEQ     10$                     ;IF TRUE, DRIVE IS NOT ONLINE, EXIT
005124  022767  000113  173166      CMP     #<ST.DRV+SC.INV>,P.STS+RSPACK   ;IS IT THE INVALID SDI RESPONCE?
```

MODULE CODE

```
1464 005132  001460                        BEQ     10$                     ;IF TRUE, DRIVE IS NOT ONLINE, EXIT
1465 005134  000461                        BR      12$                     ;ELSE HARD ERROR
1466 005136                          3$:
1467 005136  122700  000012                CMPB    @ST.CNT,R0              ; CONTROLLER ERROR?
1468 005142  001004                        BNE     4$                      ;IF NOT NEXT TEST
1469 005144  012767  000003  172734        MOV     @ERR.3,ERRTYP           ;ELSE, SET ERROR TYPE
1470 005152  000531                        BR      ERRORH                  ;AND HARD ERROR
1471 005154                          4$:
1472 005154  122700  000011                CMPB    @ST.HST,R0              ; HOST BUFFER ACCESS ERROR?
1473 005160  001004                        BNE     5$                      ;IF NOT NEXT TEST
1474 005162  012767  000032  172716        MOV     @ERR.32,ERRTYP          ;ELSE, SET ERROR TYPE
1475 005170  000522                        BR      ERRORH                  ;AND HARD ERROR
1476 005172                          5$:
1477 005172  122700  000010                CMPB    @ST.DAT,R0              ; DATA ERROR?
1478 005176  001004                        BNE     6$                      ;IF NOT NEXT TEST
1479 005200  012767  000001  172700        MOV     @ERR.1,ERRTYP           ;ELSE, SET ERROR TYPE
1480 005206  000533                        BR      ERRORS                  ;AND SOFT ERROR
1481 005210                          6$:
1482 005210  122700  000006                CMPB    @ST.WPR,R0              ; WRITE PROTECTED?
1483 005214  001431                        BEQ     12$                     ;ELSE HARD ERROR
1484 005216                          8$:
1485 005216  122700  000004                CMPB    @ST.AVL,R0              ; STILL AVAILABLE?
1486 005222  001005                        BNE     9$                      ;IF NOT NEXT TEST
1487 005224  022767  000003  173160        CMP     @OP.GUS,P.OPCD+CMPACK   ;ELSE, IF COMMAND WAS
1488                                                                        ;       GET UNIT STATUS
1489 005232  001427                        BEQ     14$                     ;THEN EXPECTED & LEAVE ROUTINE
1490 005234  000421                        BR      12$                     ;ELSE HARD ERROR
1491 005236                          9$:
1492 005236  122700  000003                CMPB    @ST.OFL,R0              ; UNIT OFFLINE?
1493 005242  001022                        BNE     13$                     ;IF NOT NEXT TEST
1494                                 ; *** OFFLINE WHEN TRIED ONLINE OR GET UNIT STATUS
1495 005244  022767  000011  173140        CMP     @OP.ONL,P.OPCD+CMPACK   ;WAS IT AN ONLINE COMMAND?
1496 005252  001410                        BEQ     10$                     ;IF SO, SET CARRY/EXIT
1497 005254  022767  000003  173130        CMP     @OP.GUS,P.OPCD+CMPACK   ;IS IT GET UNIT STATUS COMMAND?
1498 005262  001404                        BEQ     10$                     ;IF SO, SET CARRY/EXIT
1499 005264  022767  000042  173120        CMP     @OP.WR,P.OPCD+CMPACK    ;IS IT WRITE COMMAND?
1500 005272  001002                        BNE     12$                     ;IF NOT, REPORT HARD ERROR
1501 005274  000261                  10$:  SEC                             ;ELSE, SET CARRY TO
1502 005276  000207                        RTS     PC                      ;AND RETURN TO DROP DRIVE/AWAIT AVAILABLE DRIVE
1503                                 ; *** HARD ERROR EXIT WITH ERROR TYPE = 6
1504 005300                          12$:
1505 005300  012767  000006  172600        MOV     @ERR.6,ERRTYP           ;ELSE, SET ERROR TYPE
1506 005306  000453                        BR      ERRORH                  ;AND HARD ERROR
1507                                 ; *** SOFT ERROR EXIT WITH ERROR TYPE = 0
1508 005310                          13$:
1509 005310  000472                        BR      ERRORS                  ;ERROR WITH ERRTYP = 0 & IS A SOFT ERROR
1510                                                                        ;  ST.CMP,ST.MFE,.ST.ABO,ST.CMD
1511                                 ; *** SUCESSFUL EXIT
1512 005312                          14$:
1513 005312  000241                        CLC                             ;CLEAR CARRY 'CAUSE PACKET IS OK
1514 005314  000207                        RTS     PC                      ;ELSE, OK, SO FAR.
1515 005316                          15$:
1516                                 ; *** WAIT FOR ATTENTION INTERRUPT
1517                                 ; *** DID WE GET AN AVAILABLE ATTENTION MESSAGE THAT WE EXPECTED?
1518 005316  005767  173202                TST     EXPAV
```

MODULE CODE

```
1519 005322  001004                        BNE     16$                     ;IF EXPAV IS NOT 0, WE GOT ONE WE DIDN'T EXPECT
1520 005324  012767  177777  173172        MOV     #177777,EXPAV           ;CLEAR EXPECTED AVAILABLE ATTENTION MESSAGE WORD
1521 005332  000767                        BR      14$                     ; AND RETURN
1522 005334                          16$:
1523 005334  052767  140000  172720        BIS     @<RG.OWN+RG.FLG>,RSPONC+2       ;
1524 005342  000626                        BR      INTERP                  ;WAIT FOR RESPONCE OF LAST PACKET SENT
1525
```

```
                        ;*****************************************************************************
                        ;                                                                            *
                        ;       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                         *
                        ;                                                                            *
                        ;*****************************************************************************
005344                  CLRPAK:
005344  017767  172702  173142          MOV     @SAREG,NUM              ;IF SA REG NOT ZERO, STORE IN NUM
005352  001421                          BEQ     5$                      ;IF SA REG IS ZERO, CLEAR PACKETS
                                        ;*************************************************************
                                                                        ;CONVERT NUM TO ASCII AND
                                                                        ;STORE AT ADR1
005354  104420  000000' 000514'         OTOA$,BEGIN,NUM,ADR1
005362  000526'
                                        ;*************************************************************
005364  104403  000000' 006024'         MSGN$,BEGIN,SANOTO      ;ASCII MESSAGE CALL WITH COMMON HEADER
005372  010346                          MOV     R3,-(SP)                ;SAVE R3
005374  010446                          MOV     R4,-(SP)                ;SAVE R4
005376  004767  174010                  JSR     PC,INITUD               ;RE INIT SA REGISTER
005402  012603                          MOV     (SP)+,R3                ;RESTORE R3
005404  012604                          MOV     (SP)+,R4                ;RESTORE R4
005406  004767  177252                  JSR     PC,SCC                  ;SET CONTROLLER CHARS AGAIN
005412  005267  172426                  INC     HRDCNT                  ;INCREMENT HARD ERROR COUNT
                                                                        ;DOING THIS WILL CAUSE ANOTHER CALL TO CLRPAK
005416  012702  000064          5$:     MOV     #52.,R2                 ;R2 = # OF WORDS TO CLEAR
005422  012705  000302'                  MOV     #RSPLEN,R5              ;R5 -> RSPLEN, 1ST WORD TO CLEAR
005426  005025          6$:     CLR     (R5)+                   ;CLEAR WORD
005430  005302                          DEC     R2                      ;R2 = ZERO? (DONE CONDITION)
005432  001375                          BNE     6$                      ;IF NOT ZERO, LOOP
005434  000207                          RTS     PC                      ;RETURN
                        ;*****************************************************************************
                        ;                                                                            *
                        ;       HARD ERROR      CARRY WILL BE SET                                    *
                        ;                                                                            *
                        ;*****************************************************************************
005436                  ERRORH:
005436  032767  000004  172352          BIT     #SR.REP,SR1             ;DO WE REPORT THE ERROR?
005444  001403                          BEQ     7$                      ;IF SO, REPORT
005446  005267  172372                  INC     HRDCNT                  ;ELSE, INCREMENT THE HARD ERROR
                                                                        ; COUNT IF NOT REPORTED
005452  000407                          BR      8$                      ;SKIP REPORT
005454  004767  000056          7$:     JSR     PC,SETTAB               ;SET UP TABLE
                                        ;*************************************************************
005460  104405  000000' 000000          HRDER$,BEGIN,NULL       ;
                                        ;*************************************************************
005466  004767  000070                  JSR     PC,PRINTE
005472  000261          8$:     SEC
005474  000207                          RTS     PC                      ;RETURN TO CYCLED
```

```
                                ;*****************************************************************************
                                ;                                                                            *
                                ;       SOFT ERROR      CARRY WILL BE SET                                    *
                                ;                                                                            *
                                ;*****************************************************************************
ERRORS:
005476  032767  000004  172312          BIT     #SR.REP,SR1             ;DO WE REPORT THE ERROR?
005504  001403                          BEQ     9$                      ;IF SO, REPORT
005506  005267  172330                  INC     SOFCNT                  ;ELSE, INCREMENT THE HARD ERROR
                                                                        ; COUNT IF NOT REPORTED
005512  000407                          BR      10$                     ;SKIP REPORT
005514  004767  000016          9$:     JSR     PC,SETTAB               ;SET UP TABLE
                                        ;*************************************************************************
005520  104406  000000' 000000          SOFER$,BEGIN,NULL       ;
                                        ;*************************************************************************
005526  004767  000030                  JSR     PC,PRINTE
005532  000261                  10$:    SEC                             ;SET CARRY
005534  000207                          RTS     PC                      ;RETURN TO CYCLED

                                ;*****************************************************************************
                                ;                                                                            *
                                ;       SETTAB                                                               *
                                ;                                                                            *
                                ;       SET UP A TABLE OF VALUES FOR A SOFT OR HARD ERROR                    *
                                ;                                                                            *
                                ;*****************************************************************************
SETTAB:
005536  016767  172244  172334          MOV     ADDR,CSRA               ;SET UP CONTROL STATUS REG REPORT
005544  016767  172550  172332          MOV     P.STS+RSPACK,ASTAT      ;SET UP STATUS
005552  017767  172474  172322          MOV     @SAREG,ACSR             ;REPORT WHAT IS STATUS REG
005560  000207                          RTS     PC
                                ;*****************************************************************************
                                ;
                                ;       PRINT EXTENDED ERROR MESSAGE
                                ;
                                ;       PRINT STATUS, OPCODE, UNIT NUMBER, BYTE COUNT, LBN AND ADDRESS
                                ;
                                ;*****************************************************************************
PRINTE:
005562                                  ;*************************************************************************
                                        ;CONVERT P.STS+RSPACK TO ASCII AND
                                        ;STORE AT ADR1
005562  104420  000000' 000320'          OTOA$,BEGIN,P.STS+RSPACK,ADR1
005570  000526'
                                        ;*************************************************************************
                                        ;*************************************************************************
                                        ;CONVERT P.OPCD+RSPACK TO ASCII AND
                                        ;STORE AT ADR2
005572  104420  000000' 000316'          OTOA$,BEGIN,P.OPCD+RSPACK,ADR2
005600  000535'
                                        ;*************************************************************************
                                        ;*************************************************************************
                                        ;CONVERT P.UNIT+RSPACK TO ASCII AND
                                        ;STORE AT ADR3
005602  104420  000000' 000312'          OTOA$,BEGIN,P.UNIT+RSPACK,ADR3
```

```
      005610  000544'
                                        ;*****************************************************************
                                        ;*****************************************************************
1615                                    ;CONVERT P.BCNT+RSPACK TO ASCII AND
                                        ;STORE AT ADR4
      005612  104420  000000' 000322'  OTOA$,BEGIN,P.BCNT+RSPACK,ADR4
      005620  000553'
                                        ;*****************************************************************
                                        ;*****************************************************************
1616                                    ;CONVERT P.LBN+2+CMPACK TO ASCII AND
                                        ;STORE AT ADR5
      005622  104420  000000' 000440'  OTOA$,BEGIN,P.LBN+2+CMPACK,ADR5
      005630  000562'
                                        ;*****************************************************************
                                        ;*****************************************************************
1617                                    ;CONVERT P.LBN+CMPACK TO ASCII AND
                                        ;STORE AT ADR6
      005632  104420  000000' 000436'  OTOA$,BEGIN,P.LBN+CMPACK,ADR6
      005640  C00571'
                                        ;*****************************************************************
                                        ;*****************************************************************
1618                                    ;CONVERT P.ADEA+CMPACK TO ASCII AND
                                        ;STORE AT ADR7
      005642  104420  000000' 000424'  OTOA$,BEGIN,P.ADEA+CMPACK,ADR7
      005650  000600'
                                        ;*****************************************************************
                                        ;*****************************************************************
1619                                    ;CONVERT P.ADPA+CMPACK TO ASCII AND
                                        ;STORE AT ADR8
      005652  104420  000000' 000422'  OTOA$,BEGIN,P.ADPA+CMPACK,ADR8
      005660  000607'
                                        ;*****************************************************************
1620  005662  104403  000000' 006066'  MSGN$,BEGIN,BANNER      ;ASCII MESSAGE CALL WITH COMMON HEADER
1621  005670  000207                   RTS     PC
1622
```

```
                                        .SBTTL  MODULE MESSAGES
                                INITE1: MSG2
                                        MSG4
                                        177777

                                INITER: MSG3
                                        ADR1
                                        MSG10
                                        ADR2
                                        MSG14
                                        ADR3
                                        177777

                                INITE2: MSG2
                                        MSG5
                                        177777

                                INITE3: MSG2
                                        MSG6
                                        177777

                                DRP1:   MSG8
                                        ADR2
                                        MSG9
                                        MSG20
                                        ADR1
                                        MSGD1
                                        177777

                                DRP2:   MSG8
                                        ADR2
                                        MSG9
                                        MSG20
                                        ADR1
                                        MSGD2
                                        177777

                                DRP3:   MSG8
                                        ADR2
                                        MSG9
                                        MSG20
                                        ADR1
                                        MSGD3
                                        177777

                                ERRPAS: MSG11
                                        ADR2
                                        MSG12
                                        ADR3
                                        MSG13
                                        ADR1
                                        MSG1
                                        177777

                                SANOTO: MSG17
```

```
1679 006026  000526'                        ADR1
1680 006030  006632'                        MSG18
1681 006032  177777                         177777
1682
1683 006034  006524'                UNIOFF: MSG16
1684 006036  000526'                        ADR1
1685 006040  177777                         177777
1686
1687 006042  007246'                WARN1:  MSG40
1688 006044  177777                         177777
1689
1690 006046  007123'                WARN2:  MSG37
1691 006050  177777                         177777
1692
1693 006052  007054'                WARN3:  MSG36
1694 006054  177777                         177777
1695
1696 006056  006305'                ABORT:  MSG7
1697 006060  177777                         177777
1698
1699 006062  006705'                ZERO:   MSG19
1700 006064  177777                         177777
1701
1702 006066  006760'                BANNER: MSG21
1703 006070  000526'                        ADR1
1704 006072  007052'                        MSG23
1705 006074  000535'                        ADR2
1706 006076  007052'                        MSG23
1707 006100  000544'                        ADR3
1708 006102  007052'                        MSG23
1709 006104  000553'                        ADR4
1710 006106  007052'                        MSG23
1711 006110  000562'                        ADR5
1712 006112  007052'                        MSG23
1713 006114  000571'                        ADR6
1714 006116  007052'                        MSG23
1715 006120  000600'                        ADR7
1716 006122  007052'                        MSG23
1717 006124  000607'                        ADR8
1718 006126  006132'                        MSG1
1719 006130  177777                         177777
```

```
                                        .SBTTL  MORE MODULE MESSAGES
                                        .NLIST  BEX

006132      045     000         MSG1:   .ASCIZ  '%'
006134      045     103     117 MSG2:   .ASCIZ  '%CONTROLLER INIT ERROR, '
006165      045     123     101 MSG3:   .ASCIZ  '%SA REGISTER = '
006205      106     117     125 MSG4:   .ASCIZ  'FOUND BY DIAGNOSTIC '
006232      123     124     105 MSG5:   .ASCIZ  'STEP NOT SET.'
006250      105     130     120 MSG6:   .ASCIZ  'EXPECTED DATA WAS INCORRECT '
006305      045     122     105 MSG7:   .ASCIZ  '%RETRY COUNT EXCEEDED, ABORT'
006342      045     104     122 MSG8:   .ASCIZ  '%DRIVE '
006352      040     104     122 MSG9:   .ASCIZ  ' DROPPED. '
006365      040     111     116 MSG10:  .ASCIZ  ' IN STEP '
006377      045     123     117 MSG11:  .ASCIZ  '%SOFT ERROR COUNT #'
006423      040     040     040 MSG12:  .ASCIZ  '      *** HARD ERROR COUNT #'
006461      045     103     110 MSG13:  .ASCIZ  '%CHECK DATA ERROR COUNT #'
006513      045     101     104 MSG14:  .ASCIZ  '%ADDR = '
006524      045     125     116 MSG16:  .ASCIZ  '%UNIT WAS FOUND OFFLINE.  UNIT NUMBER = '
006575      045     123     101 MSG17:  .ASCIZ  '%SA REGISTER IS NOT ZERO, = '
006632      045     103     117 MSG18:  .ASCIZ  '%CONTROLLER IS GOING THROUGH INITILIZATION'
006705      045     122     111 MSG19:  .ASCIZ  '%RING AREA NOT CLEARED%'
006735      045     104     105 MSG20:  .ASCIZ  '%DEVICE ID BIT = '
006760      045     123     124 MSG21:  .ASCIZ  '%STATUS ENDCOD UNITNU BYTECO HI LBN LO LBN EXTADR PHYADR%'
                                MSG23:  .ASCIZ  ' '
007052      040     000         MSG36:  .ASCIZ  ' ! OPERATING WITH NO DISK ACCESSING !%'
007054      040     041     040 MSG37:  .ASCII  <07><07>' ! CUSTOMER DATA WILL BE OVERWRITTEN !%'
007123      007     007     040         .ASCII  '  ----------------------------------%'<07><07>
007174      040     055     055         .ASCIZ
007246      040     111     106 MSG40:  .ASCII  ' IF YOU WISH TO DESTROY CUSTOMER DATA, SET BIT1 (NOT BIT0)%'
007341      040     111     116         .ASCIZ  ' IN SWITCH REGISTER 1(SR1) OF DUBE? EQUAL TO 1.%'
007422      045     105     122 MSGD1:  .ASCIZ  '%ERRORS CAUSED DRIVE TO BE DROPPED%'
007466      045     125     116 MSGD2:  .ASCIZ  '%UNIT WAS NOT FOUND BY EXERCISER%'
007530      045     104     126 MSGD3:  .ASCIZ  '%DVID1 BIT SET HIGHER THAN ACTUAL # OF DRIVES FOUND%'
                                        .EVEN
007616                          RBUF:   .BLKW   256.                    ;THE READ BUFFER
            000001                      .END
```

Symbol table

```
ABORT   006056R
ACSR    000102R
ADDR    000006R
ADDR22= 001000
ADR1    000526R
ADR2    000535R
ADR3    000544R
ADR4    000553R
ADR5    000562R
ADR6    000571R
ADR7    000600R
ADR8    000607R
APTPRE= 000200
ASB     000106R
ASR04   002524R
ASTAT   000104R
AUTO  = 000010
AVAILB  004604R
AWAS    000110R
BANNER  006066R
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
CAPRES= 000004
CDATA$= 104412
CDERCT  000144R
CDWDCT  000146R
CF.AVL= 000200
CF.MSC= 000100
CF.OTH= 000040
CF.SMD= 000002
CF.THS= 000020
CF.576= 000001
CINTR   000254R
CKHNG$= 000001
CLKPRE= 000001
CLKSP$= 104422
CLRPAK  005344R
CMDREF  000270R
CMPACK  000402R
CMPLEN  000376R
CMPVIR  000400R
COMMND  000264R
CONFIG  000056R
CPAKEA  000464R
CPAKEP  000470R
CPAKPA  000462R
CPAKPP  000466R
CSRA    000100R
CVTADR  002032R
CYCLED  003314R
CYCLEL  003612R
DAP     004554R
DATCK$= 104411
DATER$= 104404
DOINTR  003574R
DROP1   004100R
DROP2   004110R
DROP3   004120R
DROP4   004126R
DRP1    005732R
DRP2    005750R
DRP3    005766R
DVICE   000630R
DVID1   000014R
EA      000476R
EA22    000502R
ECCMEM= 000100
EF.BBR= 000200
EF.BBU= 000100
EF.FRS= 000200
EF.LOG= 000040
EF.LST= 000100
EF.MIS= 000001
EF.SEX= 000020
ENDIT$= 104413
END$  = 104410
ERRORH  005436R
ERRORS  005476R
ERROR1  002640R
ERROR2  002636R
ERROR3  002634R
ERROR5  002622R
ERRPAS  006004R
ERRTYP  000106R
ERR.0 = 000000
ERR.1 = 000001
ERR.3 = 000003
ERR.32= 000032
ERR.6 = 000006
EXIT$ = 104400
EXPAV   000524R
FREE    000150R
GETPA$= 104415
GETWB   001722R
GTSTAA  004654R
GTSTAT  004626R
GWBUF$= 104414
HC.CCT= 000006
HC.CMD= 000004
HC.CPK= 000366R
HC.RCT= 000002
HC.RES= 000000
HC.RPK= 000306R
HC.SIZ= 000010
HRDCNT  000044R
HRDER$= 104405
HRDPAS  000050R
ICONT   000036R
ICOUNT  000040R
IDNUM   000122R
IMODX.= 000000
INDPAR= 000040
INIT    000030R
INITER  005700R
INITE1  005672R
INITE2  005716R
INITE3  005724R
INITUD  001412R
INTA    002562R
INTERP  005020R
INTR    000120R
KTPRES= 000400
KTXTND= 040000
LIMIT   000626R
LOOP1   001276R
LOOP2   001312R
L.CHVR= 000015
L.CNTI= 000014
L.CYL = 000034
L.DATA= 000050
L.ERLC= 000030
L.EVNT= 000000
L.GRP = 000040
L.SCTR= 000042
L.SLOT= 000002
L.TRCK= 000041
L.UHVR= 000027
L.UNTI= 000016
L.USVR= 000026
L.VSER= 000044
MAITP   004356R
MAITR   004252R
MAITW   004306R
MAP22$= 104416
MA10NC  001374R
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  000000R
MODSP   000252R
MSGD1   007422R
MSGD2   007466R
MSGD3   007530R
MSGN$ = 104403
MSGS$ = 104402
MSG$  = 104401
MSG1    006132R
MSG10   006365R
MSG11   006377R
MSG12   006423R
MSG13   006461R
MSG14   006513R
MSG16   006524R
MSG17   006575R
MSG18   006632R
MSG19   006705R
MSG2    006134R
MSG20   006735R
MSG21   006760R
MSG23   007052R
MSG3    006165R
MSG36   007054R
MSG37   007123R
MSG4    006205R
MSG40   007246R
MSG5    006232R
MSG6    006250R
MSG7    006305R
MSG8    006342R
MSG9    006352R
NCPUOP= 000020
NOAPTY= 000002
NTRUPT  005024R
NULL  = 000000
NUM     000514R
OLDEA   000520R
OLDPA   000516R
ONLINE  004722R
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      000474R
PARPRE= 002000
PASCNT  000034R
PA22    000500R
PDPF11= 000002
PDPLSI= 020000
PDP44 = 100000
PDP60 = 004000
PDP70 = 010000
PICKBK  003644R
PIRQ$ = 000004
PKTSIZ= 000060
POPSP = 005726
POPSP2= 022626
PORTID  000636R
PRMMS$= 000002
PRINTE  005562R
PRNMSG  000522R
PRTNUM= 000017
PRTY  = 000000
PRTY0 = 000000
PRTY1 = 000040
```

Symbol table

```
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.MSTI= 000020
P.MTMO= 000020
P.LBN = 000034
P.LGDT= 000014
P.MEDI= 000034
P.MLUN= 000014
P.MOD = 000012
P.OPCD= 000010
P.OTRF= 000014
P.RBN = 000014
P.RBNS= 000056
P.RCTC= 000057
P.RCTS= 000054
P.RGID= 000034
P.RGOF= 000040
P.SFTW= 000040

P.SHST= 000042
P.SHUN= 000040
P.STS = 000012
P.SZOF= 000012
P.TIME= 000024
P.TRCK= 000044
P.UNFL= 000016
P.UNIT= 000004
P.UNSZ= 000044
P.UNTI= 000024
P.USEF= 000022
P.VRSN= 000014
P.VSER= 000050
QMON22= 000010
RANACC  003654R
RAND8$  104417
RANNUM  000054R
RBUF    007616R
RBUFEA  000130R
RBUFEP  000506R
RBUFPA  000126R
RBUFPP  000504R
RBUFSZ  000132R
RBUFVA  000124R
READ    004462R
READA   004514R
RESTRT  001066R
RESTR1  001136R
RESTR2  001112R
RES1    000056R
RES2    000060R
RG.FLG= 040000
RG.OWN= 100000
RH70  = 001000
RINTR   000256R
RLIM  = 000004
RPAKEA  000370R
RPAKEP  000374R
RPAKPA  000366R
RPAKPP  000372R
RSPACK  000306R
RSPEA   000274R
RSPEP   000300R
RSPLEN  000302R

RSPONC  000260R
RSPPA   000272R
RSPPP   000276R
RSPVIR  000304R
RSTRT   000112R
R6    =%000006
R7    =%000007
SANOTO  006024R
SAREG   000252R
SA.CMD= 034000
SA.CME= 000360
SA.DIA= 000400
SA.ERC= 003777
SA.ERR= 100000
SA.GO = 000001
SA.INE= 000200
SA.INT= 000200
SA.LFC= 040000
SA.MAP= 000100
SA.MCV= 000377
SA.NSI= 002000
SA.PRG= 000001
SA.Q22= 001000
SA.RSE= 000017
SA.RSP= 003400
SA.SM = 000040
SA.S1 = 004000
SA.S2 = 010000
SA.S3 = 020000
SA.S4 = 040000
SA.VCE= 000177
SA.VEC= 000177
SBADR   000102R
SCC     004664R
SC.DIS= 000400
SC.DUP= 000200
SC.INV= 000100
SC.IOP= 000100
SC.NVL= 000040
SC.STO= 000040
SECH    000620R
SECL    000616R
SEND    004750R
SEQACC  004032R

SETTAB  005536R
SETUP   003010R
SNDSTP  002536R
SOFCNT  000042R
SOFER$= 104406
SOFPAS  000046R
SPOINT  000032R
SPSIZ = 000040
SR.CMP= 004000
SR.DUA= 001000
SR.REP= 000004
SR.SEQ= 002000
SR.SUM= 000010
SR.XFR= 000002
SR1     000016R
SR2     000020R
SR3     000022R
SR4     000024R
START   000710R
STAT    000026R
ST.ABO= 000002
ST.AVL= 000004
ST.CMD= 000001
ST.CMP= 000007
ST.CNT= 000012
ST.DAT= 000010
ST.DIA= 000037
ST.DRV= 000013
ST.HST= 000011
ST.MFE= 000005
ST.MSK= 000037
ST.OFL= 000003
ST.SUB= 000040
ST.SUC= 000000
ST.WPR= 000006
SVR0    000062R
SVR1    000064R
SVR2    000066R
SVR3    000070R
SVR4    000072R
SVR5    000074R
SVR6    000076R
SYSCNT  000052R

TABLEW  000644R
TEND    000704R
TIMER = 002260
TIMOUT= 005670
TRPDFD= 000023
TRY     000634R
TSTOFL  003244R
UF.CMR= 000001
UF.CMW= 000002
UF.INA= 040000
UF.RMV= 000200
UF.RPL= 010000
UF.SCH= 004000
UF.SCL= 002000
UF.WBN= 000040
UF.WPH= 020000
UF.WPS= 010000
UF.576= 000004
UNIOFF  006034R
UNITFL  000640R
UNITNO  000632R
UNSZH   000624R
UNSZL   000622R
USTACK= 000001
VA      000472R
VECTOR  000010R
WARN1   006042R
WARN2   006046R
WARN3   006052R
WASADR  000104R
WBUFEA  000136R
WBUFEP  000512R
WBUFPA  000134R
WBUFPP  000510R
WBUFRQ  000140R
WBUFSZ  000142R
WDFR    000116R
WDTO    000114R
WORK    000642R
WRITE   004426R
WRITEA  004440R
XFLAG   000005R
ZERO    006062R

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        010616    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13663 Words  ( 54 Pages)
```

DUBE DEC/X11 SYSTEM EXERCISER M MACRO V05.03 Friday 27-Sep-85 16:23 Page 28-3
Symbol table

Size of core pool: 19372 Words (74 Pages)
Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:00:49.34
XDUBE0,XDUBE0/CR/-SP=XDUBE0.DOC,DDXCOM.MAC,XDUBE0.MAC

```
SYMBOL  VALUE          REFERENCES
ABORT     006056 R      16-904    #27-1696
ACSR      000102 R     #3-3      *26-1602
ADDR      000006 R     #3-3       13-530    15-644    16-897    25-1437   26-1600
ADDR22  = 001000       #3-3       15-711    15-734    15-751    15-768    15-785
ADR1      000526 R     #3-85      14-569   *14-570    16-883    20-1229   26-1541   26-1612   27-1630   27-1649
                        27-1657   27-1665   27-1674   27-1679   27-1684   27-1703
ADR2      000535 R     #3-87      14-565   *14-566    16-881    20-1227  *20-1228   26-1613   27-1632   27-1646
                        27-1654   27-1662   27-1670   27-1705
ADR3      000544 R     #3-89      14-567   *14-568    16-897    26-1614   27-1634   27-1672   27-1707
ADR4      000553 R     #3-91      26-1615    27-1709
ADR5      000562 R     #3-93      26-1616    27-1711
ADR6      000571 R     #3-95      26-1617    27-1713
ADR7      000600 R     #3-97      26-1618    27-1715
ADR8      000607 R     #3-99      26-1619    27-1717
APTPRE  = 000200       #3-3
ASB       000106 R     #3-3
ASR04     002524 R      15-715    15-738    15-755    15-772    15-789   #15-811
ASTAT     000104 R     #3-3      *26-1601
AUTO    = 000010       #3-3
AVAILB    004604 R      18-1088  #23-1358
AWAS      000110 R     #3-3
BANNER    006066 R      26-1620  #27-1702
BEGIN     000000 R     #3-3       13-512    13-515    13-518    13-527    14-565    14-567    14-569    14-571
                        14-586    14-608    14-618    15-649    15-649    15-708    15-720    15-729    15-741
                        15-746    15-758    15-763    15-775    15-780    15-792    15-834    15-837    16-874
                        16-875    16-881    16-883    16-887    16-891    16-895    16-897    16-898    16-899
                        16-904    16-905    18-1084   18-1114   19-1132   19-1134   20-1227   20-1229   20-1233
                        20-1237   20-1239   25-1439   25-1442   26-1541   26-1542   26-1570   26-1587   26-1612
                        26-1613   26-1614   26-1615   26-1616   26-1617   26-1618   26-1619   26-1620
BIT0    = 000001       #3-3       6-213
BIT00   = 000001       #5-131
BIT01   = 000002        3-20     #5-132
BIT02   = 000004        3-21     #5-133
BIT03   = 000010        3-22     #5-134
BIT04   = 000020       #5-135
BIT05   = 000040       #5-136
BIT06   = 000100       #5-137
BIT07   = 000200       #5-138     15-669    15-671
BIT08   = 000400       #5-139
BIT09   = 001000        3-23     #5-140
BIT1    = 000002       #3-3
BIT10   = 002000       #3-3        3-24     #5-141
BIT11   = 004000       #3-3        3-25     #5-142
BIT12   = 010000       #3-3      #5-143
BIT13   = 020000       #3-3      #5-144
BIT14   = 040000       #3-3      #5-145      7-218
BIT15   = 100000       #3-3      #5-146      7-217     15-663    15-668
BIT2    = 000004       #3-3
BIT3    = 000010       #3-3
BIT4    = 000020       #3-3
```

SYMBOL	VALUE		REFERENCES								
BIT5	= 000040		#3-3								
BIT6	= 000100		#3-3								
BIT7	= 000200		#3-3								
BIT8	= 000400		#3-3								
BIT9	= 001000		#3-3								
BREAK$	= 104407		#3-3	15-649	15-649						
BR1	000012	R	#3-3	15-831							
BR2	000013	R	#3-3								
BTOD$	= 104421		#3-3	14-565	14-567	14-569	20-1227				
CAPRES	= 000004		#3-3								
CDATA$	= 104412		#3-3	18-1084	18-1114						
CDERCT	000144	R	#3-3	*13-520	14-569						
CDWDCT	000146	R	#3-3								
CF.AVL	= 000200		#10-302	24-1393							
CF.MSC	= 000100		#10-303								
CF.OTH	= 000040		#10-304								
CF.SMD	= 000002		#10-306								
CF.TMS	= 000020		#10-305								
CF.576	= 000001		#10-307								
CINTR	000254	R	#3-31								
CKHNG$	= 000001		#3-3								
CLKPRE	= 000001		#3-3								
CLKSP$	= 104422		#3-3								
CLRPAK	005344	R	21-1257	21-1277	22-1307	22-1325	23-1345	23-1358	23-1371	24-1389	24-1406
			#26-1538								
CMDREF	000270	R	#3-36	*13-526	15-688	*17-926	*25-1430	25-1432			
CMPACK	000402	R	#3-55	15-762	*21-1258	*21-1259	*21-1260	*21-1278	*21-1279	*21-1280	*21-1287
			*21-1290	*22-1308	*22-1310	*22-1311	*22-1326	*22-1328	*22-1329	*22-1333	*22-1334
			*22-1335	*23-1346	*23-1359	*23-1372	*23-1373	*24-1392	*24-1393	*24-1409	*25-1432
			*25-1433	25-1487	25-1495	25-1497	25-1499	26-1616	26-1617	26-1618	26-1619
CMPLEN	000376	R	#3-53	*21-1289	*22-1331	*23-1348	*23-1375	*24-1390	*24-1408		
CMPVIR	000400	R	#3-54	*21-1291							
COMMND	000264	R	#3-34	*15-696	*15-697	*25-1434	*25-1435				
CONFIG	000056	R	#3-3	15-711	15-734	15-751	15-768	15-785			
CPAKEA	000464	R	#3-57	*15-764							
CPAKEP	000470	R	#3-59	15-697	*15-773	*15-777					
CPAKPA	000462	R	#3-56	*15-765							
CPAKPP	000466	R	#3-58	15-696	*15-770	*15-776					
CSRA	000100	R	#3-3	*26-1600							
CVTADR	002032	R	14-550	#15-728							
CYCLED	0033[illegible]		14-597	#18-1017	18-1047						
CYCLEL	0036[illegible]		14-617	#18-1109							
DAP	0045[illegible]	R	#23-1345								
DATCK$	= 104411		#3-3								
DATER$	= 104404		#3-3								
DOINTR	003574	R	#18-1096								
DROP1	004100	R	14-599	#20-1212							
DROP2	004110	R	17-941	#20-1215							
DROP3	004120	R	17-959	#20-1218							
DROP4	004126	R	20-1214	20-1217	#20-1220						

```
SYMBOL  VALUE          REFERENCES
DRP1      005732 R      20-1233   #27-1645
DRP2      005750 R      20-1237   #27-1653
DRP3      005766 R      20-1239   #27-1661
DVICE     000630 R     #4-113     *13-523    14-583    14-592    14-605    17-968    20-1220   *20-1226
DVID1     000014 R     #3-3        13-523
EA        000476 R     #3-63      *15-719    15-730    15-737    15-747    15-754    15-764    15-771    15-781
                        15-788
EA22      000502 R     #3-65       15-722    15-743    15-760    15-777    15-794
ECCMEM  = 000100       #3-3
EF.BBR  = 000200       #9-281
EF.BBU  = 000100       #9-282
EF.FRS  = 000200       #12-408
EF.LOG  = 000040       #9-283
EF.LST  = 000100       #12-409
EF.MIS  = 000001       #12-410
EF.SEX  = 000020       #9-284
ENDIT$  = 104413       #3-3        14-608    14-618
END$    = 104410       #3-3        14-586    16-875    16-905
ERRORH    005436 R      17-997     18-1076    25-1470   25-1475   25-1506   #26-1563
ERRORS    005476 R      25-1480    25-1509   #26-1580
ERROR1    002640 R      15-654     15-841    #16-879
ERROR2    002636 R      15-846    #16-878
ERROR3    002634 R      15-658     15-672    15-678    #16-877
ERROR5    002622 R      15-687    #16-874
ERRPAS    006004 R      14-571    #27-1669
ERRTYP    000106 R     #3-3       *25-1456  *25-1469  *25-1474  *25-1479  *25-1505
ERR.0   = 000000       #5-151
ERR.1   = 000001       #5-152      25-1479
ERR.3   = 000003       #5-153      25-1469
ERR.32  = 000032       #5-155      25-1474
ERR.6   = 000006       #5-154      25-1505
EXIT$   = 104400       #3-3        15-834    25-1439
EXPAV     000524 R     #3-82      *13-521   *18-1034  *18-1097  25-1518   *25-1520
FREE      000150 R     #3-3
GETPA$  = 104415       #3-3        15-729    15-746    15-763    15-780
GETWB     001722 R      14-588     14-616   #15-708
GTSTAA    004654 R      23-1361   #23-1375
GTSTAT    004626 R      17-937     18-1029   18-1067   #23-1371
GWBUF$  = 104414       #3-3        15-708
HC.CCT  = 000006       #7-228
HC.CMD  = 000004       #7-227
HC.CPK  = 000366 R     #7-230
HC.RCT  = 000002       #7-226
HC.RES  = 000000       #7-225
HC.RPK  = 000306 R     #7-229      7-230
HC.SIZ  = 000010       #7-222
HRDCNT    000044 R     #3-3        14-567   *26-1549  *26-1566
HRDER$  = 104405       #3-3        16-898    26-1570
HRDPAS    000050 R     #3-3
ICONT     000036 R     #3-3
```

```
SYMBOL  VALUE              REFERENCES
ICOUNT    000040 R         #3-3
IDNUM     000122 R         #3-3
IMODX.  = 000000           #3-3          15-708
INDPAR  = 000040           #3-3
INIT      000030 R         #3-3
INITER    005700 R          16-899      #27-1629
INITE1    005672 R          16-887      #27-1625
INITE2    005716 R          16-891      #27-1637
INITE3    005724 R          16-895      #27-1641
INITUD    001412 R          14-557      #15-642      26-1545
INTA      002562 R          15-830      #15-836
INTERP    005020 R          18-1036      18-1099    #25-1438     25-1524
INTR      000120 R         #3-3
KTPRES  = 000400           #3-3
KTXTND  = 040000           #3-3
LIMIT     000626 R         #4-111        21-1281     21-1283
LOOP1     001276 R          14-584      #14-588      14-609
LOOP2     001312 R         #14-591       14-606
L.CHVR  = 000015           #12-418
L.CNTI  = 000014           #12-416      #12-417
L.CYL   = 000034           #12-423
L.DATA  = 000050           #12-428
L.ERLC  = 000030           #12-422
L.EVNT  = 000000           #12-414
L.GRP   = 000040           #12-424
L.SCTR  = 000042           #12-426
L.SLOT  = 000002           #12-415
L.TRCK  = 000041           #12-425
L.UHVR  = 000027           #12-421
L.UNTI  = 000016           #12-419
L.USVR  = 000026           #12-420
L.VSER  = 000044           #12-427
MAITP     004356 R          21-1262      21-1284    #21-1286
MAITR     004252 R          18-1111     #21-1257
MAITW     004306 R          18-1110     #21-1277
MAP22#  = 104416           #3-3          15-720      15-741      15-758      15-775      15-792
MA1ONC    001?74 R          14-576      #14-616      14-619
MD.CMP  = 040000           #9-264
MD.ERR  = 010000           #9-266
MD.EXP  = 100000           #9-265
MD.FEU  = 000001           #9-275
MD.NXU  = 000001           #9-277        23-1373
MD.SCH  = 004000           #9-267
MD.SCL  = 002000           #9-268
MD.SEC  = 001000           #9-269
MD.SER  = 000400           #9-270
MD.SPD  = 000001           #9-274
MD.SSH  = 000200           #9-271
MD.VOL  = 000002           #9-276
MD.WBN  = 000100           #9-272
```

```
SYMBOL  VALUE          REFERENCES
MD.WBV  = 000040       #9-273
MODNAM    000000 R     #3-3
MODSP     000252 R      3-3          #3-3
MSGD1     007422 R      27-1650      #28-1750
MSGD2     007466 R      27-1658      #28-1751
MSGD3     007530 R      27-1666      #28-1752
MSGN$   = 104403       #3-3          13-512       13-515       13-518       14-571       16-874       16-887       16-891       16-895
                        16-899       16-904       20-1233      20-1237      20-1239      26-1542      26-1620
MSGS$   = 104402       #3-3
MSG$    = 104401       #3-3
MSG1      006132 R      27-1675      27-1718      #28-1724
MSG10     006365 R      27-1631      #28-1733
MSG11     006377 R      27-1669      #28-1734
MSG12     006423 R      27-1671      #28-1735
MSG13     006461 R      27-1673      #28-1736
MSG14     006513 R      27-1633      #28-1737
MSG16     006524 R      27-1683      #28-1738
MSG17     006575 R      27-1678      #28-1739
MSG18     006632 R      27-1680      #28-1740
MSG19     006705 R      27-1699      #28-1741
MSG2      006134 R      27-1625      27-1637      27-1641      #28-1725
MSG20     006735 R      27-1648      27-1656      27-1664      #28-1742
MSG21     006760 R      27-1702      #28-1743
MSG23     007052 R      27-1704      27-1706      27-1708      27-1710      27-1712      27-1714      27-1716      #28-1744
MSG3      006165 R      27-1629      #28-1726
MSG36     007054 R      27-1693      #28-1745
MSG37     007123 R      27-1690      #28-1746
MSG4      006205 R      27-1626      #28-1727
MSG40     007246 R      27-1687      #28-1748
MSG5      006232 R      27-1638      #28-1728
MSG6      006250 R      27-1642      #28-1729
MSG7      006305 R      27-1696      #28-1730
MSG8      006342 R      27-1645      27-1653      27-1661      #28-1731
MSG9      006352 R      27-1647      27-1655      27-1663      #28-1732
NCPUOP  = 000020       #3-3
NOAPTY  = 000002       #3-3
NTRUPT    005024 R      14-572       15-701       #25-1441
NULL    = 000000       #3-3          16-898       26-1570      26-1587
NUM       000514 R      #3-72        *16-880       16-881      *16-882      16-883      *20-1213     *20-1216     *20-1219     *20-1231
                       *20-1235     *26-1539      26-1541
OLDEA     000520 R      #3-74        *13-533       14-553      *14-556
OLDPA     000516 R      #3-73        *13-532       14-551      *14-555
ONLINE    004722 R      18-1041      #24-1406
OPEN    = 000000         3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3
                         3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3
                         3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3
                         3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3          3-3
                         3-3          #3-3
OP.ABO  = 000001       #6-234
OP.ACC  = 000020       #8-235
OP.ACP  = 000102       #8-256        25-1448
```

SYMBOL	VALUE		REFERENCES								
OP.AVA	= 000100		#8-253	25-1446							
OP.AVL	= 000010		#8-236	23-1359							
OP.CCD	= 000021		#8-237								
OP.CMP	= 000040		#8-238								
OP.DAP	= 000013		#8-239	23-1346							
OP.END	= 000200		#8-252								
OP.ERL	= 000101		#8-254								
OP.ERS	= 000022		#8-240								
OP.FLU	= 000023		#8-241								
OP.GCS	= 000002		#8-242								
OP.GUS	= 000003		#8-243	23-1372	25-1487	25-1497					
OP.MRD	= 000030		#8-250	21-1258							
OP.MWR	= 000031		#8-251	21-1278							
OP.ONL	= 000011		#8-244	24-1409	25-1495						
OP.RD	= 000041		#8-245	22-1326							
OP.RPL	= 000024		#8-246								
OP.SCC	= 000004		#8-247	24-1392							
OP.SHC	= 000102		#8-255								
OP.SUC	= 000012		#8-248								
OP.WR	= 000042		#8-249	22-1308	25-1499						
OTOA$	= 104420		#3-3	16-881	16-883	16-897	20-1229	26-1541	26-1612	26-1613	26-1614
			26-1615	26-1616	26-1617	26-1618	26-1619				
PA	000474	R	#3-62	*15-718	15-720	15-731	15-736	15-741	15-748	15-753	15-758
			15-765	15-770	15-775	15-782	15-787	15-792			
PARPRE	= 002000		#3-3								
PASCNT	000034	R	#3-3	14-562							
PA22	000500	R	#3-64	15-721	15-742	15-759	15-776	15-793			
PDPF11	= 000002		#3-3								
PDPLSI	= 020000		#3-3								
PDP44	= 100000		#3-3								
PDP60	= 004000		#3-3								
PDP70	= 010000		#3-3								
PICKBK	003644	R	18-1066	#19-1128							
PIRQ$	= 000004		#3-3	15-837	25-1442						
PKTSIZ	= 000060		#7-223	7-230							
POPSP	= 005726		#3-3								
POPSP2	= 022626		#3-3								
PORTID	000636	R	#4-116	*14-594	*17-932	*17-958	20-1220	20-1222	20-1226	20-1229	
PRHMS$	= 000002		#3-3								
PRINTE	005562	R	26-1571	26-1588	#26-1611						
PRNMSG	000522	R	#3-78	*13-522	14-562	*14-564					
PRTNUM	= 000017		#3-77	3-78	13-522	14-564					
PRTY	= 000000		#3-3								
PRTY0	= 000000		3-3	#3-3							
PRTY1	= 000040		#3-3								
PRTY2	= 000100		#3-3								
PRTY3	= 000140		#3-3								
PRTY4	= 000200		3-3	#3-3							
PRTY5	= 000240		#3-3								
PRTY6	= 000300		#3-3								

```
SYMBOL  VALUE          REFERENCES
PRTY7   = 000340       #3-3
PS      = 177776       #3-3
PSW     = 177776       #3-3
PUSH    = 005746       #3-3
PUSH2   = 024646       #3-3
PWRFLG  = 000002       #3-3
P.ADEA  = 000022       #10-319   *21-1259  *21-1279  *22-1311  *22-1328   26-1618
P.ADPA  = 000020       #10-318   *21-1260  *21-1280  *22-1310  *22-1329   26-1619
P.BCNT  = 000014       #10-316   #11-356   *21-1287  *22-1333   26-1615
P.BUFF  = 000020       #10-317
P.CMST  = 000020       #11-362
P.CNCL  = 000022       #11-391
P.CNT   = 000006       #11-400
P.CNTF  = 000016       #10-339   #11-389   *24-1393
P.CNTI  = 000024       #11-392
P.CPSP  = 000042       #10-332
P.CRF   = 000000       #10-312   #11-351   #11-398   *25-1432
P.CTMO  = 000020       #11-390
P.CYL   = 000050       #11-373
P.ELGF  = 000034       #10-330
P.FBBK  = 000034       #11-357
P.FLGS  = 000011       #11-354   #11-402
P.GRP   = 000046       #11-372
P.HSTI  = 000020       #10-328   #11-367   #11-381
P.HTMO  = 000020       #10-340
P.LBN   = 000034       #10-320   *22-1334  *22-1335   26-1616   26-1617
P.LGDT  = 000014       #11-404
P.MEDI  = 000034       #11-393
P.MLUN  = 000014       #11-365   #11-379
P.MOD   = 000012       #10-315   *23-1373
P.OPCD  = 000010       #10-314   #11-353   #11-401   *21-1258  *21-1278  *22-1308  *22-1326  *23-1346  *23-1359
                       *23-1372  *24-1392  *24-1409   25-1446   25-1448   25-1487   25-1495   25-1497   25-1499
                        26-1613
P.OTRF  = 000014       #10-324   #11-361
P.RBN   = 000014       #10-335
P.RBNS  = 000056       #11-375
P.RCTC  = 000057       #11-376
P.RCTS  = 000054       #11-374
P.RGID  = 000034       #10-345   *21-1290
P.RGOF  = 000040       #10-346
P.SFTW  = 000040       #10-321   #11-358
P.SHST  = 000042       #11-370   #11-394
P.SHUN  = 000040       #10-331   #11-369   #11-383
P.STS   = 000012       #11-355    17-987    17-991   25-1452   25-1461   25-1463   26-1601   26-1612
P.SZOF  = 000012       #11-403
P.TIME  = 000024       #10-342
P.TRCK  = 000044       #11-371
P.UNFL  = 000016       #10-327   #11-366   #11-380
P.UNIT  = 000004       #10-313   #11-352   #11-399   17-943   *25-1433   26-1614
P.UNSZ  = 000044       #11-384    18-1043   18-1044
```

SYMBOL CROSS REFERENCE

SYMBOL	VALUE		REFERENCES								
P.UNIT	= 000024		#10-329	#11-368	#11-382						
P.USEF	= 000022		#10-341								
P.VRSN	= 000014		#10-338	#11-388							
P.VSER	= 000050		#11-385								
QMON22	= 000010		#3-3	15-709	15-732	15-749	15-766	15-783			
RANACC	003654	R	#19-1131								
RAND#	= 104417		#3-3	13-527	19-1132	19-1134					
RANNUM	000054	R	#3-3	13-528	19-1133	19-1135					
RBUF	007616	R	3-3	#28-1754							
RBUFEA	000130	R	#3-3	*15-747							
RBUFEP	000506	R	#3-68	*15-756	*15-760	21-1259	22-1328				
RBUFPA	000126	R	#3-3	*15-748	18-1084	18-1114					
RBUFPP	000504	R	#3-67	*15-753	*15-759	21-1260	22-1329				
RBUFSZ	000132	R	#3-3	21-1261	22-1327						
RBUFVA	000124	R	#3-3	15-745							
READ	004462	R	18-1079	#22-1325							
READA	004514	R	22-1312	#22-1330							
RESTRT	001066	R	3-3	#14-549							
RESTR1	001136	R	14-554	#14-559							
RESTR2	001112	R	14-552	#14-555							
RES1	000056	R	#3-3								
RES2	000060	R	#3-3	15-709	15-732	15-749	15-766	15-783			
RG.FLG	= 040000		#7-218	18-1035	18-1098	25-1434	25-1436	25-1523			
RG.OWN	= 100000		#7-217	18-1035	18-1098	25-1435	25-1436	25-1523			
RH70	= 001000		#3-3								
RINTR	000256	R	#3-32	*25-1445							
RLIM	= [illegible]		#4-121	16-901							
RPAKEA	[illegible]	R	#3-49	*15-781							
RPAKEP	000374	R	#3-51	15-700	*15-790	*15-794					
RPAKPA	000366	R	#3-48	*15-782							
RPAKPP	000372	R	#3-50	15-699	*15-787	*15-793					
RSPACK	000306	R	#3-47	7-229	15-779	17-943	17-987	17-991	18-1043	18-1044	25-1446
			25-1448	25-1452	25-1461	25-1463	26-1601	26-1612	26-1613	26-1614	26-1615
RSPEA	000274	R	#3-41	*15-730							
RSPEP	000300	R	#3-43	14-553	14-556	15-681	*15-739	*15-743			
RSPLEN	000302	R	#3-45	*21-1288	*22-1330	*23-1347	*23-1360	*23-1374	*24-1391	*24-1407	26-1552
RSPONC	000260	R	#3-33	15-683	*15-699	*15-700	15-728	*18-1035	*18-1098	*25-1436	*25-1523
RSPPA	000272	R	#3-40	*15-731							
RSPPP	000276	R	#3-42	14-551	14-555	15-673	*15-736	*15-742			
RSPVIR	000304	R	#3-46	*21-1292							
RSTRT	000112	R	#3-3								
R6	=%000006		#3-3								
R7	=%000007		#3-3								
SANOTO	006024	R	26-1542	#27-1678							
SAREG	000252	R	#3-27	*13-530	*13-531	15-646	15-695	15-832	15-839	16-882	26-1539
			26-1602								
SA.CMD	= 034000		#6-185								
SA.CME	= 000360		#6-200								
SA.DIA	= 000400		#6-176	15-655	15-656						
SA.ERC	= 003777		#6-170								

```
SYMBOL  VALUE          REFERENCES
SA.ERR  = 100000       #6-166      15-647      15-840
SA.GO   = 000001       #6-213      15-694
SA.INE  = 000200       #6-190
SA.INT  = 000200       #6-183      15-663
SA.LFC  = 040000       #6-205
SA.MAP  = 000100       #6-177
SA.MCV  = 000377       #6-209
SA.NSI  = 002000       #6-174
SA.PRG  = 000001       #6-195
SA.Q22  = 001000       #6-175
SA.RSE  = 000017       #6-199
SA.RSP  = 003400       #6-184
SA.SM   = 000040       #6-178
SA.S1   = 004000       #6-162      15-653      15-655      15-656      15-666
SA.S2   = 010000       #6-163      15-671
SA.S3   = 020000       #6-164
SA.S4   = 040000       #6-165
SA.VCE  = 000177       #6-189
SA.VEC  = 000177       #6-182
SBADR     000102 R     #3-3
SCC       004664 R      17-925     #24-1388    26-1548
SC.DIS  = 000400       #12-454      17-987      17-991
SC.DUP  = 000200       #12-456      17-987      17-991
SC.INV  = 000100       #12-461      25-1463
SC.IOP  = 000100       #12-453      17-987      17-991
SC.NVL  = 000040       #12-451      17-987      17-991
SC.STO  = 000040       #12-460      25-1461
SECH      000620 R     #4-106      #13-529     #19-1141    #19-1157    19-1158     #19-1161    #19-1166    #19-1187    19-1189
                      #19-1192      22-1335
SECL      000616 R     #4-105      #13-528     #19-1174    #19-1181    19-1183     #19-1191     22-1334
SEND      004750 R      21-1293     22-1336      23-1349     23-1376    24-1395     #25-1430     25-1431
SEQACC    004032 R      19-1130    #19-1180
SETTAB    005536 R      26-1569     26-1586    #26-1599
SETUP     003010 R      14-582     #17-923
SNDSTP    002536 R      15-667      15-674      15-682     #15-829
SOFCNT    000042 R     #3-3         14-565     #26-1583
SOFER#  = 104406       #3-3         26-1587
SOFPAS    000046 R     #3-3
SPOINT    000032 R     #5-3
SPSIZ   = 000040       #2-28       3-3
SR.CMP  = 004000       #3-25        18-1081     18-1112
SR.DUA  = 001000       #3-23        18-1018     18-1074     18-1085     25-1459
SR.REP  = 000004       #3-21        26-1564     26-1581
SR.SEQ  = 002000       #3-24        19-1129
SR.SUM  = 000010       #3-22        14-560
SR.XFR  = 000002       #3-20        13-511      13-513      14-575
SR1       000016 R     #3-3        #13-511      13-513      14-560      14-575      18-1018     18-1074     18-1081     18-1085
                        18-1112     19-1129     25-1459     26-1564     26-1581
SR2       000020 R     #3-3         15-690
SR3       000022 R     #3-3
```

SYMBOL VALUE REFERENCES

```
SR4       000024 R    #3-3
START     000710 R     3-3       #13-508     16-903
STAT      000026 R    #3-3
ST.ABO  = 000002      #12-436
ST.AVL  = 000004      #12-438     18-1069     25-1485
ST.CMD  = 000001      #12-435
ST.CMP  = 000007      #12-441
ST.CNT  = 000012      #12-444     25-1467
ST.DAT  = 000010      #12-442     25-1477
ST.DIA  = 000037      #12-446
ST.DRV  = 000013      #12-445     17-985      25-1457     25-1461     25-1463
ST.MST  = 000011      #12-443     25-1472
ST.MFE  = 000005      #12-439
ST.MSK  = 000037      #12-432     17-991
ST.OFL  = 000003      #12-437     17-983      25-1492
ST.SUB  = 000040      #12-433
ST.SUC  = 000000      #12-434
ST.WPR  = 000006      #12-440     25-1482
SVR0      000062 R    #3-3
SVR1      000064 R    #3-3
SVR2      000066 R    #3-3
SVR3      000070 R    #3-3
SVR4      000072 R    #3-3
SVR5      000074 R    #3-3
SVR6      000076 R    #3-3
SYSCNT    000052 R    #3-3
TABLEW    000644 R    #4-123      #13-524     #13-525     14-589      17-929      17-946
TEND      000704 R    #4-126       14-603      17-948      17-972
TIMER   = 002260      #3-80        15-645
TIMOUT  = 005670      #4-120
TRPDFD  = 000023      #3-3         3-3         3-3         #3-3        3-3         #3-3        3-3         3-3
                      #3-3         3-3         3-3         #3-3        3-3         #3-3        3-3         3-3
                      #3-3         3-3         3-3         #3-3        3-3         #3-3        3-3         3-3
                      #3-3         3-3         3-3         #3-3        3-3         #3-3        3-3         3-3
                      #3-3         3-3         3-3         #3-3        3-3         #3-3        3-3         3-3
                      #3-3         3-3         3-3         #3-3
                      #3-3         3-3         3-3         #3-3        3-3         3-3         #3-3        3-3         3-3
TRY       000634 R    #4-115      #14-558     #15-702     #16-900      16-901      18-1021     #18-1049
TSTOFL    003244 R    #17-983      18-1031
UF.CMR  = 000001      #9-288
UF.CMW  = 000002      #9-289
UF.INA  = 040000      #9-291
UF.RMV  = 000200      #9-292
UF.RPL  = 010000      #9-290
UF.SCH  = 004000      #9-293
UF.SCL  = 002000      #9-294
UF.WBN  = 000040      #9-295
UF.WPH  = 020000      #9-296
UF.WPS  = 010000      #9-297
UF.576  = 000004      #9-298
```

SYMBOL	VALUE		REFERENCES								
UNIOFF	006034	R	#27-1683								
UNITFL	000640	R	#4-117								
UNITNO	000632	R	#4-114	*14-574	*14-595	*17-930	17-931	*17-943	17-952	*17-957	17-963
			17-965	*17-970	20-1227	25-1433					
UNSZH	000624	R	#4-109	*18-1043	18-1046	19-1142	19-1145	19-1158	19-1189		
UNSZL	000622	R	#4-108	*18-1044	*18-1064	19-1167	19-1169	19-1171	19-1183		
USTACK	= 000001		#3-3								
VA	000472	R	#3-61	*15-728	15-729	*15-745	15-746	*15-762	15-763	*15-779	15-780
VECTOR	000010	R	#3-3	14-572	15-660	15-701	15-829				
WARN1	006042	R	13-512	#27-1687							
WARN2	006046	R	13-515	#27-1690							
WARN3	006052	R	13-518	#27-1693							
WASADR	000104	R	#3-3								
WBUFEA	000136	R	#3-3	15-714	15-719						
WBUFEP	000512	R	#3-70	*15-716	*15-722	21-1279	22-1311				
WBUFPA	000134	R	#3-3	15-713	15-718						
WBUFPP	000510	R	#3-69	*15-713	*15-721	21-1280	22-1310				
WBUFRQ	000140	R	#3-3								
WBUFSZ	000142	R	#3-3	18-1059	21-1281	21-1285	22-1309				
WDFR	000116	R	#3-3								
WDTO	000114	R	#3-3								
WORK	000642	R	#4-118	*17-936	*17-939						
WRITE	004426	R	18-1072	#22-1307							
WRITEA	004440	R	#22-1309								
XFLAG	000005	R	#3-3								
ZERO	006062	R	16-874	#27-1699							

MACRO CROSS REFERENCE

MACRO NAME	REFERENCES
BKMOD	#2-124
BREAK	#2-222 15-649
BTOD	#2-246 14-565 14-567 14-569 20-1227
CKDATA	#2-282 18-1084 18-1114
CLKSP	#2-149
DATACK	#2-291
DATERR	#2-175
DFSEVN	#2-314 3-3 3-3 3-3 3-3 3-3 3-3 3-3 3-3 3-3 3-3 3-3
DSEVNT	#2-324 3-3 3-3 3-3 3-3 3-3 3-3 3-3 3-3
END	#2-212 16-875
ENDIT	#2-203 14-608 14-618
ENDMOD	#2-208 14-586 16-905
EQUATS	#2-330 3-3
EXIT	#2-257 15-834 25-1439
GETPA	#2-273 15-729 15-746 15-763 15-780
GMBUFF	#2-261 15-709
HLFBRK	#2-227
HRDER	#2-165 16-898 26-1570
IOMOD	#2-120
IOMODP	#2-144
IOMODR	#2-140
IOMODX	#2-136 3-3
MAP22	#2-277 15-720 15-741 15-758 15-775 15-792
MODULE	#2-29 3-3
MSG	#2-191
MSGN	#2-195 13-512 13-515 13-518 14-571 16-874 16-887 16-891 16-895 16-899 16-904 20-1233 20-1237 20-1239 26-1542 26-1620
MSGS	#2-199
NBKMOD	#2-132
OTOA	#2-232 16-881 16-883 16-897 20-1229 26-1541 26-1612 26-1613 26-1614 26-1615 26-1616 26-1617 26-1618 26-1619
PIRQ	#2-216 15-837 25-1442
RAND	#2-161 13-527 19-1132 19-1134
SBKMOD	#2-128
SOFER	#2-181 26-1587