RA80 UDA & DISK DRV DIAG AH-S831C-MC
CZUDCCO FICHE 1 OF 2

RA80 UDA & DISK DRV DIAG AH-S831C-MC
CZUDCCO FICHE 2 OF 2

digital

```
.REM *
```

IDENTIFICATION

PRODUCT CODE: AC-S830C-MC

PRODUCT NAME: CZUDCCO UDA50 & DISK DRV DIAG

PRODUCT DATE: 27-AUG-82

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: MATT TEDONE


```
.REM *
.REM &
```

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

This is the only diagnostic program provided for testing the UDA-50 Unibus Disk Controller and the disk drives connected to it. There are four tests within this diagnostic:

Test # 1 - Unibus Addressing Test. Runs the UDA-50 ROM resident diagnostics, then further tests the Unibus address interface.

Test # 2 - Disk Resident Diagnostic Test. Executes the diagnostics in each disk drive.

Test # 3 - Disk Function Test. Functionally tests each disk drive to ensure the disk can seek, read, write and format.

Test # 4 - Disk Exerciser. Exercises the disk drives in a manner similar to normal operating systems. This test should be used to gain confidence in the reliability of the disk drive.

This program is designed to handle all future disk drives that are attached to the UDA-50 without modifying or rereleasing. This is possible because the disk drives are programmed to tell this diagnostic about all their characteristics that make them different from other drives, such as number of cylinders, sectors per cylinder, etc.

Two other PDP-11 diagnostic programs are provided for the UDA-50 disk subsystem:

CZUDEC0 - UDA-50 Disk Drive Formatter.

CXUDFB0 - UDA-50 Disk Drive Formatter Data File

DEC/X11 - Unibus Exerciser can be run on the UDA-50 using the UDA-50 module DUBC0.

This diagnostic has been written for use with the Diagnostic Runtime Services Software (Supervisor). These services provide the interface to the operator and to the software environment. For a complete description of the Runtime Services, refer to the XXDP+ User's Manual. There is a brief description of the Runtime Services in section 2 of this document.

Two versions of the UDA-50 have been distributed, one in the modules M7161 and M7162, the other in modules M7485 and M7486. This diagnostic will test with versions in any combination except as stated in section 1.2. Whenever a fault is detected in a UDA-50 and the fault can be isolated to one of the two modules in the UDA-50. The two versions of that module are printed. Replace the module that exists in your UDA-50.

1.2 SYSTEM REQUIREMENTS

This program was designed using the PDP-11 Diagnostic Runtime Services revision C. Run time environments are determined by the Runtime Services and may change as new versions of the Services are developed. This program requires the following:

PDP-11 Unibus processor
28K words of memory (minimum)
Console terminal
XXDP+ load media containing this program and the ZUDDC0.PAK data file
One or more UDA-50 subsystems
Line clock - either Type L or P

Any combination of the two versions of UDA-50s (M7161-2 or M7485-6 module sets) can be tested except when all of the following is true:

1) XXDP+ load media is on a disk drive connected to a UDA-50 of the type M7485-6 modules.
2) The UDA-50 that is connected to the load media is under test.
3) At least one disk attached to a UDA-50 version M7161-2 is included in the testing.

In this case, error message number 00009 will be printed in test 4. None of the UDA-50's with the M7464-5 modules will be tested.

The line clock is used for all timed loops in the program. The diagnostic will run on a system with no clock but will hang whenever an event for which the program is waiting does not happen (i.e., a time-out error message will not result).

This diagnostic program requires that the data file ZUDDC0.PAK be on the XXDP+ system device. This data file is ordered under the name CZUDDC0. The XXDP+ system device must remain on-line during the execution of this diagnostic.

2.0 OPERATING INSTRUCTIONS

This section contains a brief description of the Runtime Services. For detailed information, refer to the XXDP+ User's Manual (CHQUS).

2.1 COMMANDS

There are eleven legal commands for the Diagnostic Runtime Services (Supervisor). This section lists the commands and gives a very brief description of them. The XXDP+ User's Manual has more details.

COMMAND	EFFECT
START	Start the diagnostic from an initial state
RESTART	Start the diagnostic without initializing
CONTINUE	Continue at test that was interrupted (after ^C)
PROCEED	Continue from an error halt
EXIT	Return to XXDP+ Monitor (XXDP+ OPERATION ONLY!)
ADD	Activate a unit for testing (all units are considered to be active at start time
DROP	Deactivate a unit
PRINT	Print statistical information (see section 4.0)
DISPLAY	Type a list of all device information
FLAGS	Type the state of all flags (see section 2.3)
ZFLAGS	Clear all flags (see section 2.3)

A command can be recognized by the first three characters. So you may, for example, type "STA" instead of "START".

2.2 SWITCHES

There are several switches which are used to modify supervisor operation. These switches are appended to the legal commands. All of the legal switches are tabulated below with a brief description of each. In the descriptions below, a decimal number is designated by "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	Execute only those tests specified in the list. List is a string of test numbers, for example - /TESTS:1:5:7-10. This list will cause tests 1,5,7,8,9,10 to be run. All other tests will not be run.
/PASS:DDDDD	Execute DDDDD passes (DDDDD = 1 to 64000)
/FLAGS:FLGS	Set specified flags. Flags are described in section 2.3.
/EOP:DDDDD	Report end of pass message after every DDDDD passes only. (DDDDD = 1 to 64000)
/UNITS:LIST	TEST/ADD/DROP only those units specified in the list. List example - /UNITS:0:5:10-12 use units 0,5,10,11,12 (unit numbers = 0-63).

Example of switch usage:

```
START/TESTS:1-5/PASS:1000/EOP:100
```

The effect of this command will be: 1) tests 1 through 5 will be executed, 2) all units will tested 1000 times and 3) the end of pass messages will be printed after each 100 passes only. A switch can be recognized by the first three characters. You may, for example, type "/TES:1-5" instead of "/TESTS:1-5".

Below is a table that specifies which switches can be used by each command.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

Flags are used to set up certain operational parameters such as looping on error. All flags are cleared at startup and remain cleared until explicitly set using the flags switch. Flags are also cleared after a START or RESTART command unless set using the flag switch. The ZFLAGS command may also be used to clear all flags. With the exception of the START, the RESTART and ZFLAGS commands, no commands affect the state of the flags; they remain set or cleared as specified by the last flag switch.

FLAG	EFFECT
HOE	Halt on error - control is returned to runtime services command mode
LOE	Loop on error
IER*	Inhibit all error reports
IBE*	Inhibit all error reports except first level (first level contains error type, number, PC, test and unit)
IXE*	Inhibit extended error reports (those called by PRINTX macro's)
PRI	Direct messages to line printer
PNT	Print test number as test executes
BOE	"BELL" on error
UAM	Unattended mode (no manual intervention)
ISR	Inhibit statistical reports
IDU	Inhibit program dropping of units
LOT	Loop on test

*Error messages are described in section 3.1

See the XXDP+ User's Manual for more details on flags. You may specify more than one flag with the FLAG switch. For example, to cause the program to loop on error, inhibit error reports and type a "BELL" on error, you may use the following string:

```
/FLAGS:LOE:IER:BOE
```

2.4 HARDWARE QUESTIONS

When a diagnostic is STARTed, the Runtime Services will prompt the user for hardware information by typing "CHANGE HW (L) ?" . When you answer this question with a "Y", the Runtime Services will ask for the number of units (in decimal). You will then be asked the following questions for each unit. When you answer this question with an "N", the Runtime Services will use the answers built into the program by the SETUP utility (see chapter 6 of the XXDP+ User's Manual). If you have never run the SETUP utility on this program file, the default values listed below (just before the question mark) will be used.

UNIBUS ADDRESS OF UDA (O) 172150 ?

Answer with the address of the UDAIP register of one UDA as addressed by the processor with memory management turned off (i.e., an even 16-bit address in the range of 160000 to 177774).

VECTOR (O) 154 ?

Answer with the interrupt vector address of the UDA. A vector address in the range of 4 to 774 may be specified. The UDA does not have a vector "hard wired" to it, so any vector not being used by this program and XXDP+ may be used.

BR LEVEL (D) 5 ?

Answer with the interrupt priority used by the UDA. Levels 4 to 7 are accepted. This level must match the level "hard wired" in the UDA by the priority plug.

UNIBUS BURST RATE (D) 63 ?

The UDA allows the ability to control the maximum number of words transferred across the UNIBUS each time the UDA becomes master. The default answer of 63 will allow for the fastest execution of this diagnostic program. You may answer with the value your operating system uses or use zero which will tell the UDA to supply a value that should work on any system. A decimal number in the range of 0 to 63 may be specified and all values should work on any system. A larger value will allow for a faster running program. The value will be passed directly to the UDA during initialization.

DRIVE NUMBER (D) 0 ?

Answer with the drive number of the drive you wish to test. This is the number which appears on the "unit plug" on the front of the disk drive. On a multi-unit drive, each sub-unit number on the drive must be tested as a separate unit to completely test the drive. A maximum of eight logical drives may be tested on one UDA at a time (UDA configuration limit).

EXERCISE ON CUSTOMER DATA AREA IN TEST 4 (L) N ?

Answer "N" to have test 4 (drive exerciser) run on the diagnostic area of the disk. Answer "Y" to run on the customer data area. A "Y" answer will destroy any customer data that may be on the disk. A warning message will be printed before testing begins if this question is answered "Y".

CUSTOMER DATA WILL BE DESTROYED ON:

UNIT	UDA AT	DRIVE
xx	xxxxxx	xxx

Unless the diagnostic is being run in unattended mode (i.e., START/FLAG:UAM command), a confirmation will also be required as follows:

ARE YOU SURE CUSTOMER DATA CAN BE DESTROYED (L) ?

If the above question is answered "N", the entire diagnostic will stop and the Runtime Services prompt will be displayed. No default answer is provided for this question.

2.5 SOFTWARE QUESTIONS

After you have answered the hardware questions or after a RESTART or CONTINUE command, the Runtire Services will ask for software parameters. You will be prompted by "CHANGE SW (L) ?" If you wish to change any parameters, answer by typing "Y". The software questions and the default values are described in the next paragraphs.

ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS (L) N ?

Tests 2 and 4 have manual intervention modes which allow additional parameters to be input to alter the normal testing of a disk drive. This question should normally be answered "N" when this diagnostic is first run. Then, depending on the errors detected, it may be desirable to change this answer to "Y" and alter the testing to further isolate the problem. If this question is answered "Y", and the UAM (unattended mode operation) flag is set, tests 2 and 4 will print a warning message that the mode cannot be entered and will proceed as if answered "N". See the description of the individual tests in section 5 for more information.

REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY

This informational message is printed to describe the use of the remaining questions. If test 4 is not being run, a "CONTROL Z" can be typed to bypass them.

ERROR LIMIT (D) 32 ?

Enter the number of hard errors allowed before a drive is dropped from exercise by test #4. A number in the range of 1 to 65535 will be accepted.

READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT (D) 0 ?

When the specified number of bytes have been read from a drive by test #4, the drive will be dropped from testing. When all drives are dropped, an end of pass will be indicated and the selected tests will be run again. This is the method used to determine how long test #4 is to run. Answer with a zero to prevent test from ending. The only other way test #4 can end is to have all drives dropped because the error limit on each is exceeded. Of course, the operator can always stop test #4 by typing a control-C.

SUPPRESS PRINTING SOFT ERRORS (L) Y?

When test #4 needs to perform retries, soft error reports will be printed to give as much information as possible. These actions are considered normal operation and are not error conditions until the retries fail. When the test is being run only to see how reliable the drive performs, this question should be answered "Y" so they are not confused with hard errors. The number of these soft errors is always reported in the statistical report. Answer "N" to see all the soft error reports.

DO INITIAL WRITE ON START (L) Y ?

If test #4 is to do data compares, the drive will need to be written with data patterns readable by the program.

If the diagnostic area is selected for testing, the initial write is always performed (regardless of how this question is answered).

If the customer data area is selected for testing, the initial write will be performed when all of the following are true:

1. This question is answered "Y".
2. This is the first time test #4 is being run after a START command.
3. The disk is write enabled.

Answering this question "N" when testing on the customer data area will normally result in data comparison errors if the disk was not previously written by this diagnostic or the formatter.

Note that write checks are not performed during the initial write.

ENABLE ERROR LOG (L) N ?

A "Y" answer will cause error messages in test #4 to be stored in a log buffer. Once the log buffer is full, additional error information is lost. The contents of the log buffer will be printed when test #4 is stopped and a statistical report requested. This log feature is intended to allow the Digital Diagnosis Center (DDC) to start test #4 then hang up from the system and let it run for some period of time. DDC can call the system back later, type control-C, then CONT and see the errors that have occurred (up to the limit of the log buffer). A message will be printed to indicate no errors have occurred if the log buffer is empty. Test #4 will not be allowed to end while the error log is enabled until the error log is printed. The log buffer will hold 30 error messages when one disk unit is being tested. The log buffer will decrease in size as more units are tested.

2.6 EXTENDED P-TABLE DIALOGUE

When you answer the hardware questions, you are building entries in a table that describes the devices under test. The simplest way to build this table is to answer all questions for each unit to be tested. If you have a multiplexed device such as a mass storage controller with several drives or a communication device with several lines, this becomes tedious since most of the answers are repetitious.

To illustrate a more efficient method, suppose you are testing a fictional device, the XY11. Suppose this device consists of a control module with eight units (sub-devices) attached to it. These units are described by the octal numbers 0 through 7. There is one hardware parameter that can vary among units called the Q-factor. This Q-factor may be 0 or 1. Below is a simple way to build a table for one XY11 with eight units.

```
# UNITS (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0<CR>
Q-FACTOR (O) 0 ?  1<CR>

UNIT 2
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  1<CR>
Q-FACTOR (O) 1 ?  0<CR>

UNIT 3
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  2<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 4
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  3<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 5
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  4<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 6
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  5<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6<CR>
Q-FACTOR (O) 0 ?  1<CR>
```

```
UNIT 8
CSR ADDRESS (O)  160000<CR>
SUB-DEVICE # (O) ?  7<CR>
Q-FACTOR (O) 1 ?  <CR>
```

Notice that the default value for the Q-factor changes when a non-default response is given. Be careful when specifying multiple units!

As you can see from the above example, the hardware parameters do not vary significantly from unit to unit. The procedure shown is not very efficient.

The Runtime Services can take multiple unit specifications however. Let's build the same table using the multiple specification feature.

```
# UNITS (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0,1<CR>
Q-FACTOR (O) 0 ?  1,0<CR>

UNIT 3
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  2-5<CR>
Q-FACTOR (O) 0 ?  0<CR>

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6,7<CR>
Q-FACTOR (O) 0 ?  1<CR>
```

As you can see in the above dialogue, the runtime services will build as many entries as it can with the information given in any one pass through the questions. In the first pass, two entries are built since two sub-devices and q-factors were specified. The Services assume that the CSR address is 160000 for both since it was specified only once. In the second pass, four entries were built. This is because four sub-devices were specified. The "-" construct tells the Runtime Services to increment the data from the first number to the second. In this case, sub-devices 2, 3, 4 and 5 were specified. (If the sub-device were specified by addresses, the increment would be by 2 since addresses must be on an even boundary.) The CSR addresses and Q-factors for the four entries are assumed to be 160000 and 0 respectively since they were only specified once. The last two units are specified in the third pass.

The whole process could have been accomplished in one pass as shown below.

```
# UNITS (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0-7<CR>
Q-FACTOR (O) 0 ?  0,1,0,,,,1,1<CR>
```

As you can see from this example, null replies (commas enclosing a null field) tell the Runtime Services to repeat the last reply.

2.7 QUICK START-UP PROCEDURE

- To start-up this program:

  1. Boot XXDP+
  2. Give the date and answer the LSI and 50HZ (if there is a clock) questions
  3. Type "R ZUDCCO"
  4. Type "START"
  5. Answer the "CHANGE HW" question with "Y"
  6. Answer all the hardware questions
  7. Answer the "CHANGE SW" question with "N"

When you follow this procedure you will be using only the defaults for flags and software parameters. These defaults are described in sections 2.3 and 2.5.

Sample of terminal dialogue to test two disks on one UDA-50:

```
DR>STA/FLA:PNT

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
UNIBUS ADDRESS OF UDA (O)  172150 ?
VECTOR (O)  154 ?
BR LEVEL (D)  5 ?
UNIBUS BURST RATE (D) ?
DRIVE NUMBER (D) 0,1
EXERCISE ON CUSTOMER DATA AREA IN TEST 4 (L) N ?

CHANGE SW (L)  ? N

TST: 001
TESTING INTERRUPT ABILITY OF UDA AT ADR 172150 VEC 154...COMPLETED

TST: 002
TST: 003
TST: 004

UNIT 0 UDA AT 172150 DRIVE 0   RUNTIME 0:02:43
INITIAL WRITE COMPLETE

UNIT 1 UDA AT 172150 DRIVE 1   RUNTIME 0:05:31
INITIAL WRITE COMPLETE

TEST 4 IN PROGRESS.    RUNTIME 0:15:00

UNIT DRIVE      SERIAL-NUMBER SEEKS MBYTES  MBYTES  HARD    SOFT    ECC
                              X1000 READ    WRITTEN ERRORS  ERRORS
   0    0                     0     3       9       6       0       0       0
   1    1                     1     3       8       6       0       0       0
```

Sample of terminal dialogue going through software questions to specify transfer limit (one disk being tested).

```
DR>STA/FLA:PNT

CHANGE HW (L)  ? N

CHANGE SW (L)  ? Y

ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS (L) N ?

REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY

ERROR LIMIT (D)  32 ?
READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT (D)  0 ? 5
SUPPRESS PRINTING SOFT ERRORS (L) Y ?
DO INITIAL WRITE ON START (L) Y ?
ENABLE ERROR LOG (L) N ?

TST: 001
TESTING INTERRUPT ABILITY OF UDA AT ADR 172150 VEC 154...COMPLETED

TST: 002
TST: 003
TST: 004

UNIT 0 UDA AT 172150 DRIVE 0   RUNTIME 0:02:43
INITIAL WRITE COMPLETE

UNIT 0 UDA AT 172150 DRIVE 0   RUNTIME 0:09:41
REACHED TRANSFER LIMIT - TESTING STOPPED

TEST 4 IN PROGRESS.    RUNTIME 0:09:41

UNIT DRIVE      SERIAL-NUMBER SEEKS MBYTES  MBYTES  HARD   SOFT    ECC
                              X1000  READ  WRITTEN ERRORS ERRORS
   0     0                     0     2       5       4      0       0     0

CZUDCC EOP     1
    0 CUMULATIVE ERRORS
TST: 001
TESTING INTERRUPT ABILITY OF UDA AT ADR 172150 VEC 154...COMPLETED

TST: 002
.
.
.
```

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

There are three levels of error messages that may be issued by a diagnostic: general, basic and extended. General error messages are always printed unless the "IER" flag is set (section 2.3). The general error message is of the form:

```
NAME  TYPE  NUMBER  ON UNIT NUMBER  TST NUMBER PC:XXXXXX
error message
```

where: NAME = diagnostic name
TYPE = error type (SYS FTL ERR, DEV FTL ERR, HRD ERR or SFT ERR)
NUMBER = error number
UNIT NUMBER = 0 - N (N is last unit in PTABLE)
TST NUMBER = test and subtest where error occurred
PC:XXXXXX = address of error message call

System fatal errors (SYS FTL ERR) are used to report errors that are fatal to the entire diagnostic program. The diagnostic stops and the Runtime Services prompt is printed.

Device fatal errors (DVC FTL ERR) are used to report errors that are fatal to the device (may be either a UDA-50 or disk drive). Testing stops on that device for the remainder of the current test.

Hard errors (HRD ERR) reports most of the errors detected. Testing will normally continue after the printing of the error.

Soft errors (SFT ERR) are used only in test 4. They present information about an error for which recovery will be attempted. These are printed only if the SUPPRESS PRINTING SOFT ERRORS software question is answered 'N' and are used only to provide a greater detail of information. During the error recovery attempt, several soft errors may be printed. Unless the soft errors are followed by a hard error message, the error condition was corrected and testing proceeds.

Basic error messages are messages that contain some additional information about the error. These are always printed unless the "IER" or "IBE" flags are set (section 2.3). These messages are printed after the associated general message.

Extended error messages contain supplementary error information such as register contents or good/bad data. These are always printed unless the "IER", "IBE" or "IXE" flags are set (section 2.3). These messages are printed after the associated general error message and any associated basic error messages.

The general and basic error messages from this diagnostic are always one line each. The basic message defines what program detected the error, the drive being tested and the time of the error.

The PDP-11 program that is loaded into memory when you give the "R ZUDCC0" command to the XXDP+ monitor is only a small part of this diagnostic. A data file called ZUDDC0.PAK on the system load device (the same device from which the "R" command read the PDP-11 program) contains four programs which are read from the file and loaded into the UDA-50 for execution. These programs are called "diagnostic machine" or DM programs. The "diagnostic machine" is the facility in the UDA-50 which executes a PDP-11 like pogram. The large majority of the testing is done by these four "diagnostic machine" programs. Once the PDP-11 program has loaded and started the "diagnostic machine" program, all it does is respond to requests from that program. These requests include such things as telling the "diagnostic machine" which disks on that UDA-50 are to be tested, printing an error message and updating statistics which are printed in the statistical report (see section 4.0).

The basic message (the second line of every error message) will be one of the following:

HOST PROGRAM UDA AT xxxxxx RUNTIME hhh:mm:ss

The host program (PDP-11) detected the error. UDA AT xxxxx identifies the address of the UDA-50 being tested. It may be omitted if the error is not specific to one UDA-50.

UNIBUS ADDRESSING DM PC:xxxx UDA AT xxxxxx RUNTIME hhh:mm:ss

The "diagnostic machine" program loaded in test 1 detected the error. DM PC xxxx identifies the address in the "diagnostic machine" program where the error message is reported.

DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss

The "diagnostic machine" program loaded in test 2 detected the error. DM PC xxxx identifies the address in the "diagnostic machine" program where the error message is reported. DRIVE xxx identifies the drive number.

DISK FUNCTIONAL DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss

The "diagnostic machine" program loaded in test 3 detected the error.

DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss

The "diagnostic machine" program loaded in test 4 detected the error.

Sample error message:

```
CZUDC DVC FTL ERR  00021 ON UNIT 00 TST 001 SUB 003 PC: xxxxxx      -  general message
HOST PROGRAM  UDA AT 172150  RUNTIME 0:00:12                        -  basic message
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE                          \
  UDASA CONTAINS 104041                                             |-  extended message
REPLACE UDA MODULE M7161 OR M7485                                  /
```

Informational messages are also printed by this program. They are usually one or two lines in length. They are printed as extended messages and are always printed unless the "IER", "IBE" or "IXE" flags are set.

Sample informational message:

```
UNIT 0  UDA AT 172150  DRIVE 0  RUNTIME 0:02:43
INITIAL WRITE COMPLETE
```

3.2 SPECIFIC ERROR MESSAGES

Following is a list of the error messages that may be printed by the diagnostic program. In the list, some of the numbers that may vary with execution or program version are shown as "xxx". These include program counters and runtime. Other numbers, such as unit number, drive number, UDA-50 address and data in registers are filled with sample numbers. Additional information about the error may follow the error message.

3.2.1 HOST PROGRAM ERROR MESSAGES (00001 to 00999)

00001 CZUDC SYS FTL ERR 00001 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS
UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE

When the hardware questions were answered, two units were selected with the same UNIBUS address but with a different vector, BR level or burst rate. A single UDA-50 can have only one vector, BR level or burst rate. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00002 CZUDC SYS FTL ERR 00002 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS
TWO UNITS SELECT THE SAME DRIVE

The hardware questions for two units were exactly the same. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00003 CZUDC SYS FTL ERR 00003 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS
MORE THAN EIGHT DRIVES SELECTED ON THIS UDA

Up to four physical disk drives can be attached to a UDA-50 at one time. A physical disk drive may be from one to four logical disk drives. Each logical disk drive is considered one unit to the diagnostic program. Even though more than eight logical disk drives can be attached to one UDA-50, the UDA-50 only supports eight. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00004 CZUDC SYS FTL ERR 00004 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM RUNTIME x:xx:xx
NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED
PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME

This program does not limit the number of units that can be tested by specifying a maximum number. What limits the number is the amount of memory used to store data on each unit. You have exceeded the number of units that are testable at one time. Start program over and select fewer units.

00005 CZUDC SYS FTL ERR 00005 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM RUNTIME x:xx:xx
CHECKSUM ERROR IN DM PROGRAM FILE

As a DM program is read from the load media, a checksum is calculated. If the checksum contained in the file does not match what is calculated, an error reading the data file is declared. Restore the data file ZUDDCO.PAK to your load media.

00006 CZUDC SYS FTL ERR 00006 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM RUNTIME x:xx:xx
TABLE INCONSISTANCY ERROR. PLEASE RE-LOAD PROGRAM

When the host program is started, controller tables are set according to the P-tables. Error 00006 will occur if the tables were corrupted after restarting the diagnostic. Load and start your program again.

00007 CZUDC SYS FTL ERR 00007 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM RUNTIME x:xx:xx
ERROR IN DM PROGRAM FILE. DM PROGRAM NOT FOUND

The host program was not able to read the DM program from the load media properly. Restore the data file ZUDDCO.PAK to your load media.

00008 CZUDC SYS FTL ERR 00008 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS
TWO UDA'S USE THE SAME VECTOR

The hardware questions for two units specified different UDA-50 Unibus addresses but identical vector addresses. The program is aborted and returns to the Runtime Services prompt so that you can change the hardware questions.

00009 CZUDC DVC FTL ERR 00009 ON UNIT 00 TST 004 SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
ILLEGAL CONFIGURATION FOR TEST 4.
CANNOT TEST ALL UDA-50S.

The Test 4 DM program was read into host memory from a load device that is attached to an enhanced UDA50 (M7485 and M7486 board set). If the enhanced UDA50 is loaded with this DM program then no other program can be read from the load media. Thus, any old UDA50's (M7161 and M7162 board set) will not be tested. This can only occur during Test 4 where both types of DM programs cannot reside in memory simultaneously. Either use another type of load media or run Test 4 twice (once on the UDA-50 attached to the load device, second on the rest of the UDA-50's).

00010 CZUDC DVC FTL ERR 00020 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
WRONG APT DIAGNOSTIC IS BEING USED WITH THIS CONTROLLER
USE CIUDx

The APT diagnostics are designed to run with one type of UDA-50 board set (either M7161-2 or M7485-6). For example, If the user is running CIUDA with a UDA-50 M7485-6 type, this error will occur. In that case the user will be told to use CIUDF. The following is a detailed description of which test is used with what configuration.

CIUDA - UDA-50 with M7161-2 modules runs tests 1-3
CIUDB - UDA-50 with M7161-2 modules runs test 4
CIUDC - UDA-50 with M7161-2 modules runs tests 1-3
CIUDD - UDA-50 with M7161-2 modules runs test 4
CIUDF - UDA-50 with M7485-6 modules runs tests 1-3
CIUDG - UDA-50 with M7485-6 modules runs test 4
CIUDH - UDA-50 with M7485-6 modules runs tests 1-3
CIUDI - UDA-50 with M7485-6 modules runs test 4

00013 CZUDC DVC FTL ERR 00013 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
MICROCODE REPORTED CONTROLLER MODEL WHICH DID NOT MATCH
GET DUST STATUS RESPONSE. TESTING ON THIS CONTROLLER STOPS.

The controller model is a four bit field (bits 7-4) in step 4 of the initialization protocol. It indicates which type of UDA-50 (either M7161-2 boards or M7485-6 boards). Another indicator of UDA-50 types is in the GET DUST STATUS response. These values are different for the two types of UDA-50s. If they do not indicate similar UDA-50 types, this error is presented. Testing stops on this controller, because it is not known which DM program to load.

00020 CZUDC DVC FTL ERR 00020 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
MEMORY ERROR TRYING TO READ UDA REGISTERS
CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161 OR M7485
OR UNIBUS
OR REPLACE UDA MODULE M7161 OR M7485

A non-existant memory error occurred when the host program tried to access the UDAIP and UDASA registers. The UDA is at another address (check the UNIBUS selection switches) or module M7161 or M7485 is broken or the UNIBUS is broken.

00021 CZUDC DVC FTL ERR 00021 ON UNIT 00 TST 001 SUB 003 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE
UDASA CONTAINS 105154
REPLACE UDA MODULE M7162 OR M7486

The UDA Resident diagnostic detected a failure. The error is displayed in the UDASA. Here are the possible error values and their meaning:

104000 - Fatal sequncer error
104040 - D processor ALU error
104041 - D proc ROM parity error
105102 - D proc with no Board #2 or RAM parity error
105105 - D proc RAM buffer error
105152 - D proc SDI error
105153 - D proc write mode wrap SERDES error
105154 - D proc read mode SERDES, RSGEN, and ECC error
106040 - U proc ALU error
106041 - U proc Control Register error
106042 - U proc DFAIL/ROM parity error/Board #1 test count is wrong
106047 - U proc Constant ROM error with D proc running SDI test
106055 - Unexpectant trap found, aborted diagnostic
106071 - U proc ROM error
106072 - U proc ROM parity error
106200 - Step 1 data error (MSB not set)
107103 - U proc RAM parity error
107107 - U proc RAM buffer error
107115 - Board #2 test count was wrong
112300 - Step 2 error
122240 - NPR error
122300 - Step 3 error
142300 - Step 4 error

Replace the board specified. M7161 and M7485 are the Unibus interface board. M7162 and M7486 are the SDI interface board. Module M7161 only works with module M7162. The same for modules M7485 and M7486. Do not intermix the two boards. It will not work!

```
00022   CZUDC DVC FTL ERR  00022 ON UNIT 00 TST 001 SUB 003 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION
        STEP BIT EXPECTED 004000
          UDASA CONTAINS  000000
        REPLACE UDA MODULE M7161 OR M7485
```

The UDA did not respond as expected during the initialization sequence which communicates using data in the UDASA register. A normal response from the UDA contains either a STEP bit or an ERROR bit defined as follows:

Bit 15 (100000)	Error bit
Bit 14 (040000)	Step 4 bit
Bit 13 (020000)	Step 3 bit
Bit 12 (010000)	Step 2 bit
bit 11 (004000)	Step 1 bit

The expected step bit nor the error bit set within the expected time.

```
00023   CZUDC DVC FTL ERR  00023 ON UNIT 00 TST 001 SUB 005 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION
        6 WORDS WERE TO BE CLEARED STARTING AT ADDRESS 040644
        FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):
              ADDRESS    CONTENTS
               040644     000010
               040650     000010
               040652     000010
        REPLACE UDA MODULE M7161 OR M7485
```

The UDA is to clear the ring structure (a communications area used by the UDA to talk to the host) in host memory before Step 4 of initialization. If the UDA diagnostics did not clear memory and did not flag an error, then error message 00023 is displayed. The contents of each word in memory is set to 177777 before the test. Failure of the UDA to clear each word indicates a fault in the address interface to the Unibus.

00024 CZUDC DVC FTL ERR 00024 ON UNIT 00 TST 001 SUB 006 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION
PURGE/POLE DIAGNOSTICS WERE REQUESTED
UDASA CONTENTS 004400

For better testing, the host can test the PURGE and POLE mechanism of the UDA. To do so the host sets bit15 of the step 3 data and sends the data to the UDA. The UDA must go to zero and wait for the purge and pole. If the UDA never went to zero, then error message 00024 is displayed. The UDA may have a bad M7161 or M7485 module or the UNIBUS maybe broken.

00025 CZUDC DVC FTL ERR 00025 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION
UDASA EXPECTED 004400
UDASA CONTAINS 004000
REPLACE UDA MODULE M7161 OR M7485

For each step of initialization, specific data is expected to be displayed in the UDASA. If the UDASA does not match the expected data, then error message 00025 is displayed. Replace UDA module M7161 or M7485.

00026 CZUDC DVC FTL ERR 00026 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST
DATA SENT TO UDASA 000001
RECEIVED FROM UDASA 000000
REPLACE UDA MODULE M7161 OR M7485

The UDA can be put into a mode where the UDASA acts as a wrap port. While the UDA is in this mode, any data being sent to the UDASA will be displayed in the UDASA within a small period of time. If the data in the UDASA does not match the data that was sent to the UDASA, then error message 00026 is displayed. Replace UDA module M7161 or M7485.

00027 CZUDC DVC FTL ERR 00027 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDASA REGISTER DID NOT CHANGE AFTER WRITING TO IT
IN PORT LOOP DIAGNOSTIC
UDASA CONTAINS 004400
REPLACE UDA MODULE M7161 OR M7485

The UDA can be put into a mode where the UDASA acts as a wrap port. While the UDA is in this mode, any data being sent to the UDASA will be displayed in the UDASA within a small period of time. After the host program sent data to it while it was in diagnostic wrap mode, the UDA did not change the contents of the UDASA. Error message 00027 is displayed. Replace UDA module M7161 or M7485.

00028 CZUDC DVC FTL ERR 00028 ON UNIT 00 TST 001 SUB 004 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA DID NOT INTERRUPT THE PDP-11
REPLACE UDA MODULE M7161 OR M7485

The host program timed out while waiting for an interrupt that had to occur. The UDA was told to use interrupts during the initialization process. The UDA then waited for the interrupt but it did not occur. Replace the UDA module M7161 or M7485.

00029 CZUDC DVC FTL ERR 00029 ON UNIT 00 TST 001 SUB 004 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA INTERRUPTED AT DIFFERENT BR LEVEL THAN SPECIFIED IN HARDWARE QUESTIONS. INTERRUPT WAS AT BR LEVEL 5
CHECK PRIORITY PLUG ON UDA MODULE M7161 OR M7485
OR CHANGE HARDWARE QUESTIONS

The priority plug on the UDA and the BR LEVEL specified during the hardware questions do not match. Either change the plug number or reanswer the hardware question. If all these have been done and there is still a problem replace UDA module M7161 or M7485.

00030 CZUDC DVC FTL ERR 00030 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM
UDASA CONTAINS 100004

A message from the UDA firmware reports an unexpected failure. An error code is presented in the UDASA. Here is a list of the codes and their meanings:

004400 - UDA has been inited by either a bus init or by writing into the UDAIP.
100001 - UNIBUS envelope/packet read error (parity or timeout)
100002 - UNIBUS envelope/packet write error (parity or timeout)
100003 - UDA ROM and RAM parity error
100004 - UDA RAM parity error
100005 - UDA ROM parity error
100006 - UNIBUS ring read error
100007 - UNIBUS ring write error
100010 - UNIBUS interrupt master failure
100011 - Host access timeout error
100012 - Host exceeded credit limit
100013 - UDA SDI hardware fatal error
100014 - DM XFC fatal error
100015 - Hardware timeout of instruction loop
100016 - Invalid virtual circuit identifier
100017 - Interrupt write error on UNIBUS

```
00031   CZUDC DVC FTL ERR  00031 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES
        ASSUME PROGRAM IS HUNG
```

All DM programs are required to communicate with the host program; so as to assure the host program that the DM program is not hung up or in an endless loop. If the DM program has not done so, the host program assumes the DM is hung and this message appears.

```
00032   CZUDC DVC FTL ERR  00032 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER
        MESSAGE BUFFER CONTAINS:
           000001 000002 000003 000004 000005 000006 000007
           000008 000009 000010 000011 000012 000013 000014
           000015 000016 000017 000018 000019 000020 000021
           000022 000023 000024 000025 000026 000027 000028
           000029 000030 000031 000032 000033 000034 000035
```

The DM program and the host program communicate with each other using packets. Each packet must have a request number set up by the DM program and interpreted by the host program. This request number is not a known request number. The problem may be the UNIBUS or either one of the UDA modules or a corrupted DM program. Word 1 contains the DM request number, and word 2 typically contains the drive number. The rest of the buffer contains information specific to a DM request. The numbers in the example show the order in which words are displayed.

```
00033   CZUDC DVC FTL ERR  00033 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        RESPONSE PACKET FROM UDA DOES NOT CONTAIN EXPECTED DATA
        EITHER UDA RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED CORRECTLY
           COMMAND PACKET SENT      RESPONSE PACKET RECEIVED
              000000 000020            000000 000020
              000000 000000            000000 000000
              000000 000002            000000 000202
              000000 014336            000000 014336
              000000 034674            000000 034674
              000000 000000            000000 000000
              000000 000000            000000 000000
              000000 051232            000000 051232
              000000 000000            000000 000000
              000000 000000            000000 000000
              000000 000000            000000 000000
```

The host program inspected the response packet which was given by to UDA. The response packet may have been in error with one of the following points:

1) The end code was not as expected.
2) The status code showed an error occurred with the last command.
3) The command reference numbers (the first word) did not match.

If 1 or 3 occurred, there may have been a transmission problem between the UDA and the host program. If 2 occurred, check the error code in the MSCP specification for further information. The packets are displayed two words per line, low order word and byte to the right (corresponding to the MSCP long-word entity).

```
00036   CZUDC DVC FTL ERR  00036 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
        HOST PROGRAM  UDA AT 172150  RUNTIME x:xx:xx
        NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS
        WHILE LOADING DM PROGRAM
```

After a DM program has been sent to the UDA, the host program expects an interrupt within 30 seconds. The interrupt is used to assure the host program that the DM program is sane. If no interrupt occurred, then error message 00036 is displayed and the DM program is assumed to be hung.

00037 CZUDC DVC FTL ERR 00037 ON UNIT 00 TST xxx SUB 000 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM
UDASA CONTAINS 100004
REPLACE UDA MODULE M7161 OR M7485

While loading the DM program to the UDA, the UDASA became non-zero. When this occurs, it signifies that the UDA microcode has run across a fatal error. The displayed value is in octal. Check the error code with the list included with error number 00030.

00038 CZUDC DVC FTL ERR 00038 ON UNIT 00 TST 001 SUB 002 PC: xxxxxx
HOST PROGRAM UDA AT 172150 RUNTIME x:xx:xx
MEMORY ERROR TRYING TO READ UDA REGISTERS
CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161 OR M7486
OR UNIBUS
OR REPLACE UDA MODULE M7161 OR M7485

A non-existant memory error occurred when the host program tried to access the UDAIP and UDASA registers while in subtest 2 of test 1. The UDA is at another address (check the UNIBUS selection switches) or module M7161 or module M7485 is broken or the UNIBUS is broken.

3.2.2 TEST 1 ERROR MESSAGES (01000 TO 01999)

01000 CZUDC HRD ERR 01000 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
UNIBUS ADDRESSING DM PC:xxxx UDA AT xxxxxx RUNTIME hhh:mm:ss
NON-EXISTANT MEMORY ERROR TRYING TO READ FROM UNIBUS.

	OCTAL	HEX
ADDRESS	000000	00000

The host has given the DM routine the range of accessible host memory. While reading one location within the range, it appeared non-existant to the UDA. Since everything within the bounds were believed to be accessible this error message will be printed. The message prints the address in octal and hex.

01001 CZUDC HRD ERR 01001 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
UNIBUS ADDRESSING DM PC:xxxx UDA AT xxxxxx RUNTIME hhh:mm:ss
PARITY ERROR ON READ FROM UNIBUS.

	OCTAL	HEX
ADDRESS	000000	00000
DATA READ	000000	0000
DATA EXPECTED	000000	0000

The host has given the DM routine the range of accessible host memory. While reading one location within the range, the DM routine has found a location with bad parity. Every location was accessed by the host program. The host program filled a location with its address. The message prints the address, the data it actually received, and the expected data it should have received in octal and hex.

01002 CZUDC HRD ERR 01002 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
UNIBUS ADDRESSING DM PC:xxxx UDA AT xxxxxx RUNTIME hhh:mm:ss
UNIBUS ADDRESSING ERROR - INCORRECT DATA READ.
MEMORY LOCATION SHOULD CONTAIN OWN ADDRESS.

	OCTAL	HEX
DATA READ	000000	0000
DATA EXPECTED	000000	0000

The host has given the DM routine the locations of accessible host memory. Every location was accessed by the host program. The host program filled a location with its address. The DM program read from one location and found that the data it read was not equal to its address. The message prints the address, the data it actually received, and the expected data it should have received in octal and hex.

01003 CZUDC HRD ERR 01003 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
UNIBUS ADDRESSING DM PC:xxxx UDA AT xxxxxx RUNTIME hhh:mm:ss
NON-EXISTANT MEMORY ERROR TRYING TO READ FROM UNIBUS WITHIN BUFFER.

	OCTAL	HEX
STARTING ADDESS OF BUFFER	123456	0A72E
BUFFER SIZE	001234	029C

After reading every accessible location of host memory, the DM routine breaks up memory into buffers. The DM routine writes and reads data patterns from each host buffer into its DM buffer. While reading one of these buffers, a non-existant memory error occurred. The message prints out the starting address of the buffer and the size of the buffer in octal(for PDP-11 users) and in hex(for VAX users) so the user can determine about where the non-existant memory location occurred.

01004 CZUDC HRD ERR 01004 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
UNIBUS ADDRESSING DM PC:xxxx UDA AT xxxxxx RUNTIME hhh:mm:ss
PARITY ERROR ON READ FROM UNIBUS WITHIN BUFFER.

	OCTAL	HEX
STARTING ADDESS OF BUFFER	123456	0A72E
BUFFER SIZE	001234	029C

After reading every accessible location of host memory, the DM routine breaks up memory into buffers. The DM routine writes and reads data patterns from each host buffer into its DM buffer. While reading one of these buffers, a parity error occurred. The message prints out the starting address of the buffer and the size of the buffer in octal(for PDP-11 users) and in hex(for VAX users) so the user can determine about where the non-existant memory location occurred.

```
01005   CZUDC HRD ERR  01005 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
        UNIBUS ADDRESSING DM PC:xxxx  UDA AT xxxxxx  RUNTIME hhh:mm:ss
        DATA COMPARE FAILED AFTER WRITE THEN READ FROM UNIBUS.
         BUFFER SIZE = 005302(O)     0AC2(X)     2754.(D)
         STARTING ADDRESSES OF BUFFERS
              OCTAL          HEX
             044232         0489A
             057056         05E2E
             071676         073BE
             104512         0894A
         CURRENT DATA PATTERN READ                    0
         LAST PATTERN WRITTEN                         0
         STARTING ADDRESS OF LAST BUFFER WRITTEN   104512(O)   0894A(X)
         NUMBER OF ERRORS FOUND                    2754.(D)
            LOCATION        DATA EXPECTED      DATA RECEIVED
          OCTAL   HEX         OCTAL   HEX       OCTAL   HEX
         057056  05E2E        111111  9249      002472  053A
         057060  05E30        044444  4924      005302  0AC2
         057062  05E32        022222  2492      000000  0000
```

After reading an entire buffer, the DM program checks each location. If any or all of the locations did not contain the expected data, this message appears. It contains the buffer size in octal, hex and decimal. The reason it appears in decimal is so the user can corralate this value with the number of errors which is printed in decimal. The starting addresses of the buffers are printed in octal and hex. There will always be at least two buffers and up to four buffers printed. The current data pattern read is printed. DM program will be testing the buffer with this data pattern. The last data pattern written by the DM program is printed. The address of the last buffer written is printed in octal and hex. As many as three errors are presented in the message. This portion presents the location of the error, the expected data and the actual data all in octal and hex.

```
01006   CZUDC HRD ERR  01006 ON UNIT 00 TST 001 SUB 007 PC: xxxxxx
        UNIBUS ADDRESSING DM PC:xxxx  UDA AT xxxxxx  RUNTIME hhh:mm:ss
        UNIBUS ADDRESSING ERROR. TWO ADDRESSES READ SAME LOCATION.
                                        OCTAL       HEX
                KNOWN GOOD ADDRESS      625252      32AAA
                ERROR ADDRESS           425252      22AAA
                ADDRESS BIT IN ERROR    200000      10000

                The UDA can only write to a small portion of memory
                because there is a PDP-11 program running in the
                memory. To verify it can address all of memory, it
                uses one location that it is permitted to write which
                it calls a "known good address". By changing only one
                bit in the address of this location it selects a "test
                address". Different patterns are written to the "known
                good address", each followed by a read of the "test
                address". If the data read from the "test address"
                matches the data written to the "known good address"
                each time, the address line is determined to be stuck.
                The "test address" is printed as the error address.
```

3.2.3 TEST 2 INFORMATIONAL MESSAGES

```
        UNIT x UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
        INFORMATION SENT BACK FROM THE DRIVE IS BEING PRESENTED.
          TEST NUMBER  0000
          DRIVE TYPE   00
          ERROR NUMBER 0000
        data

                There is not error, but it is a message.  The disk
                drive wanted the let the host know what had
                happened when the drive's internal diagnostic
                was run.  The format follows that of hard error 2021.

        UNIT x UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
        FOLLOWING REPORT HAS BEEN TRUNCATED DUE TO SIZE

                This is a message that may appear if the disk drive
                gave too much data for the DM program to handle.
                This message may preceed the previous message and
                hard error 2021.
```

3.2.4 TEST 2 ERROR MESSAGES (02000 TO 02999)

02000 CZUDC HRD ERR 02000 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
HOST SPECIFIED UNIT #0 THAT CAN'T BE FOUND.
TEST2 RESTARING

When test 2 starts executing out of the DM, it doesn't know if it had been started to execute drive diagnostics or restarted to down line load a diagnostic into the drive. If it had been restarted for the latter reason, the host must tell Test 2 which drive was to receive the diagnostic. If the drive specified by the host is not attached to the UDA or could not be located by Test 2, this error message will be printed.

02001 CZUDC HRD ERR 02001 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
CANNOT RECEIVE VALID DRIVE STATE FROM DRIVE AFTER DRIVE WAS INITED
CHECK IF DRIVE IS POWERED ON.

This error message is presented if valid drive state was not received from the drive after the drive was inited. There are two types of invalid states: no clocks or 'hard' errors. If after getting state and no clocks occur, error 2001 is reported. There may be a bad transmitter on the drive side or a bad receiver on the UDA side or the SDI cable may have taken a hit.

02002 CZUDC HRD ERR 02002 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
DRIVE STATE RECEIVED HAS BAD PARITY AFTER DRIVE WAS INITED

This error message is presented if bad parity was received from the drive after the drive was inited. There may be a bad transmitter on the drive side or a bad receiver on the UDA side or the SDI cable may have taken a hit.

02003 CZUDC HRD ERR 02003 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
DRIVE IS NOT ASSERTING RECEIVER READY IN DRIVE STATE AFTER DRIVE WAS INITED

This error message is presented if receiver ready was not received from the drive after the drive was inited. There may be a bad transmitter on the drive side or a bad receiver on the UDA side or the SDI cable may have taken a hit.

```
02004   CZUDC HRD ERR  02004 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        TIME-OUT ON SEND OF ECHO COMMAND TO DRIVE
          ECHO DATA  FF

                This error message is presented if a send of the
                ECHO command timed out.  This may be caused by
                receiver ready being deasserted.  The echo data
                is presented in hex.

02005   CZUDC HRD ERR  02005 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ERROR DURING RECEIVE OF ECHO RESPONSE FROM DRIVE
          ECHO DATA  FF

                This error message is presented if a receive of an ECHO
                command was in error.  The echo data is presented in hex.
                There  may be a bad transmitter on the drive side or a
                bad receiver on the UDA side or the SDI cable may have
                taken a hit.

02006   CZUDC HRD ERR  02006 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ECHO COMMAND RESPONDED WITH DIFFERENT DATA
          ECHO DATA SENT      00FE
          ECHO DATA RECEIVED  00FF

                This error message is presented if the data returning from
                an ECHO command did not match the data it was suppose to.
                The data presented is in hex.

02007   CZUDC HRD ERR  02007 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ERROR BIT SET IN GET STATUS RESPONSE AFTER DRIVE CLEAR COMMAND
          GET STATUS RESPONSE
        REAL TIME STATE state
        STATUS (FROM R TO L): word6  word5  word4  word3  word2  word1  word0:

                This error message is presented when an error bit
                is set in the status of a drive after the drive
                was cleared of all errors.  The data displayed
                is the responce from a GET STATUS command.
                The error bits in the responce are in bit position 3, 5
                and 6 of word2.  For further description of the
                GET STATUS responce, refer to the SDI Functional
                Spec v3.6 and the drive's functional spec.

        REAL TIME STATE state:  REAL TIME STATE 0003
                The real time state is the real time drive state <<AFTER>> Test 2
                detected the error.  <<THIS VALUE IS DISPLAYED IN HEX>>.  In this
                example, receiver ready and attention are both asserted.

                The bit positions are defined as follows:
                        0001 - Receiver ready (Test 2 able to transmit to drive)
                        0002 - Attention (error occurred or online timeout expired)
                        0040 - Available (drive offline and usuable)
                        1000 - Read/Write ready
```

The complete meaning of these bits is beyond the scope of this text, please refer to the operator documentation for the drive you are working on.

STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0:
The status is the response to the SDI GET STATUS command. These words are printed in HEX. <<NOTE THAT THE STATUS IS PRINTED OUT FROM RIGHT TO LEFT!!>>. The status' meaning is beyond the scope of this text, please refer to the operator documentation for the drive you are working on.

02008 CZUDC HRD ERR 02008 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF ONLINE COMMAND TO DRIVE

The ONLINE command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

02009 CZUDC HRD ERR 02009 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF ONLINE RESPONSE FROM DRIVE
explanation

This error message is presented if a receive of an ONLINE command was in error. An explanation of what the error was is also presented. These explanations are:

TIMEOUT ERROR OCCURED DURING RECEIVE XFC
- This error is a failure of the drive to respond to an SDI level 2 command (see the SDI specification) before the drive-supplied command timeout expires.

1ST WORD NOT START FRAME DURING RECEIVE XFC
- The first word received by the UDA from the drive was not a valid message start frame.

FRAMING ERROR OCCURED ON SDI LEVEL 0 READ DURING RECEIVE XFC
- This is caused by one of the following conditions: 1) Illegal frame code -- the frame is not a message start, continue, or end frame. 2) Illegal sequence of frames -- such as a message start frame without ever receiving a message end frame. This can be caused by the drive sending a response before the UDA asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or UDA receiver.

CHECKSUM ERROR OCCURED ON SDI LEVEL 0 READ DURING RECEIVE XFC
- The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad UDA receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable.

BUFFER SIZE SMALLER THEN RESPONSE DURING RECEIVE XFC
- A buffer size size set aside for the response was not large enough for the response received. This is caused by the drive sending a response that is incorrect for the request sent to the drive, or the drive sending some garbage with the response.

CODE FROM RECEIVE XFC WAS UNINTELLIGIBLE FROM SUBSYSTEM 0000
- The responce from the drive was not anything that was expected. Possible UDA microcode change without test 2 update.

02010 CZUDC HRD ERR 02010 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ONLINE COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The ONLINE command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status. The drive did not asseri the RECEIVER READY signal over the SDI.

02011 CZUDC HRD ERR 02011 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ONLINE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
EXPECTED RESPONSE 7E
ACTUAL RESPONSE 00

The ONLINE command did not return an expected response code. If there were at least an UNSUCCESSFUL response, test 2 will report the drive state and status. The expected response and actual response are in hex.

02012 CZUDC HRD ERR 02012 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF GET UNIT CHARACTERISTICS COMMAND TO DRIVE

The GET UNIT CHARACTERISTICS command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

02013 CZUDC HRD ERR 02013 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF GET UNIT CHARACTERISTICS COMMAND FROM DRIVE
explanation

This error message is presented if a receive of a GET UNIT CHARACTERISTICS command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

02014 CZUDC HRD ERR 02014 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
GET UNIT CHARACTERISTICS COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The GET UNIT CHARACTERISTICS command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

02015 CZUDC HRD ERR 02015 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
GET UNIT CHARACTERISTICS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
EXPECTED RESPONSE 78
ACTUAL RESPONSE 00

The GET UNIT CHARACTERISTICS command did not return an expected response code. The expected response and actual response are in hex.

02016 CZUDC HRD ERR 02016 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
HOST PROGRAM GAVE DM CODE IMPROPER DATA
EXPECTED VALUE SHOULD BE BETWEEN 0 AND 3
ACTUAL VALUE WAS xx

The host tells the DM program what to do after the DM program is done testing the drive's diagnostic. If the value is not within the expected range, this error message is printed. There is no drive problem. The problem is between the host and the UDA.

02017 CZUDC HRD ERR 02017 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF DIAGNOSE COMMAND TO DRIVE

The DIAGNOSE command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

02018 CZUDC HRD ERR 02018 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF DIAGNOSE RESPONSE FROM DRIVE
explanation

This error message is presented if a receive of a DIAGNOSE command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

```
02019   CZUDC HRD ERR  02019 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DIAGNOSE COMMAND WAS UNSUCCESSFUL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

              The DIAGNOSE command was not successful.  The drive's status
              is displayed.  See hard error 2007 for further information
              on the format of the status.

02020   CZUDC HRD ERR  02020 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DIAGNOSE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
          EXPECTED RESPONSE  FC
          ACTUAL RESPONSE    00

              The DIAGNOSE command did not return an expected
              response code.  The expected response and actual response
              are in hex.

02021   CZUDC HRD ERR  02021 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DRIVE DIAGNOSTIC REPORTS A HARD ERROR
          TEST NUMBER  0000
          DRIVE TYPE   00
          ERROR NUMBER 0000
        data
              The drive diagnostic found an error and is reporting the
              error back to the host.  All values are in hex.  TEST NUMBER
              shows what test was run.  DRIVE TYPE shows what type of
              drive was being tested.  ERROR NUMBER shows the result of
              the test.  The drive may pass back data to the host.
              This data will be presented in a 32 bit hex format following
              the error message.  More data may follow the 32 bit hex
              values.  This data is printed in ascii format.  For
              definitions of what these values mean, refer to the
              drive functional spec.

02022   CZUDC HRD ERR  02022 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        HOST PROGRAM DOWN LINE LOADED A DIAGNOSTIC WITH A ZERO BYTE COUNT

              The host program was attempting to down line load a
              diagnostic of zero length.  The DM program must have
              the byte count specified by the host.

02023   CZUDC HRD ERR  02023 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DIAGNOSTIC filnam REQUESTED BY THE DRIVE COULD NOT BE SUPPLIED BY HOST.

              The host program could not supply the diagnostic 'filnam'
              to down line load to the drive.
```

02024 CZUDC HRD ERR 02024 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF MEMORY READ COMMAND TO DRIVE

The MEMORY READ command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

02025 CZUDC HRD ERR 02025 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF MEMORY READ RESPONSE FROM DRIVE
explanation

This error message is presented if a receive of a MEMORY READ command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

02026 CZUDC HRD ERR 02026 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
MEMORY READ COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The MEMORY READ command was not successful. The drive's status is displayed. See hard error 2007 for further information on the format of the status.

02027 CZUDC HRD ERR 02027 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
MEMORY READ COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
EXPECTED RESPONSE 72
ACTUAL RESPONSE 00

The MEMORY READ command did not return an expected response code. The expected response and actual response are in hex.

02028 CZUDC HRD ERR 02028 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF MEMORY WRITE COMMAND TO DRIVE

The MEMORY WRITE command timed out while it was sent to the drive. The drive did not assert the RECEIVER READY signal over the SDI.

02029 CZUDC HRD ERR 02029 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK RESIDENT DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF MEMORY WRITE RESPONSE FROM DRIVE
explanation

This error message is presented if a receive of a MEMORY WRITE command was in error. An explanation of what the error was is also presented. These explanations are described in hard error 2009.

```
02030   CZUDC HRD ERR  02030 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        MEMORY WRITE COMMAND WAS UNSUCCESSFUL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

              The MEMORY WRITE command was not successful.  The drive's status
              is displayed.  See hard error 2007 for further information
              on the format of the status.

02031   CZUDC HRD ERR  02031 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        MEMORY WRITE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
          EXPECTED RESPONSE  7E
          ACTUAL RESPONSE    00

              The MEMORY WRITE command did not return an expected
              response code.  The expected response and actual response
              are in hex.

02032   CZUDC HRD ERR  02032 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        TIME-OUT ON SEND OF RUN COMMAND TO DRIVE

              The RUN command timed out while it was sent
              to the drive.  The drive did not assert
              the RECEIVER READY signal over the SDI.

02033   CZUDC HRD ERR  02033 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ERROR DURING RECEIVE OF RUN RESPONSE FROM DRIVE
        explanation

              This error message is presented if a receive of a RUN
              command was in error.  An explanation of what the error was
              is also presented.  These explanations are described in
              hard error 2009.

02034   CZUDC HRD ERR  02034 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        RUN COMMAND WAS UNSUCCESSFUL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

              The RUN command was not successful.  The drive's status
              is displayed.  See hard error 2007 for further information
              on the format of the status.
```

```
02035   CZUDC HRD ERR  02035 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        RUN COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
          EXPECTED RESPONSE  7E
          ACTUAL RESPONSE    00

                The RUN command did not return an expected
                response code.  The expected response and actual response
                are in hex.

02036   CZUDC HRD ERR  02036 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        TIME-OUT ON SEND OF RECALIBRATE COMMAND TO DRIVE

                The RECALIBRATE command timed out while it was sent
                to the drive.  The drive did not assert
                the RECEIVER READY signal over the SDI.


02037   CZUDC HRD ERR  02037 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ERROR DURING RECEIVE OF RECALIBRATE RESPONSE FROM DRIVE
        explanation

                This error message is presented if a receive of a RECALIBRATE
                command was in error.  An explanation of what the error was
                is also presented.  These explanations are described in
                hard error 2009.

02038   CZUDC HRD ERR  02038 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        RECALIBRATE COMMAND WAS UNSUCCESSFUL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The RECALIBRATE command was not successful.  The drive's status
                is displayed.  See hard error 2007 for further information
                on the format of the status.

02039   CZUDC HRD ERR  02039 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        RECALIBRATE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
          EXPECTED RESPONSE  7E
          ACTUAL RESPONSE    00

                The RECLAIBRATE command did not return an expected
                response code.  The expected response and actual response
                are in hex.

02040   CZUDC HRD ERR  02040 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        TIME-OUT ON SEND OF GET STATUS COMMAND TO DRIVE

                The GET STATUS command timed out while it was sent
                to the drive.  The drive did not assert
                the RECEIVER READY signal over the SDI.
```

```
02041   CZUDC HRD ERR  02041 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ERROR DURING RECEIVE OF GET STATUS RESPONSE FROM DRIVE
        explanation

                This error message is presented if a receive of a GET STATUS
                command was in error.  An explanation of what the error was
                is also presented.  These explanations are described in
                hard error 2009.

02042   CZUDC HRD ERR  02042 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        GET STATUS COMMAND WAS UNSUCCESSFUL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The GET STAUTS command was not successful.  The drive's status
                is displayed.  See hard error 2007 for further information
                on the format of the status.

02043   CZUDC HRD ERR  02043 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        GET STATUS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
          EXPECTED RESPONSE  F6
          ACTUAL RESPONSE    00

                The GET STATUS command did not return an expected
                response code.  The expected response and actual response
                are in hex.

02044   CZUDC HRD ERR  02044 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        TIME-OUT ON SEND OF DRIVE CLEAR COMMAND TO DRIVE

                The DRIVE CLEAR command timed out while it was sent
                to the drive.  The drive did not assert
                the RECEIVER READY signal over the SDI.

02045   CZUDC HRD ERR  02045 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        ERROR DURING RECEIVE OF DRIVE CLEAR RESPONSE FROM DRIVE
        explanation

                This error message is presented if a receive of a DRIVE CLEAR
                command was in error.  An explanation of what the error was
                is also presented.  These explanations are described in
                hard error 2009.
```

```
02046   CZUDC HRD ERR  02046 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DRIVE CLEAR COMMAND WAS UNSUCCESSFUL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

              The DRIVE CLEAR command was not successful.  The drive's status
              is displayed.  See hard error 2007 for further information
              on the format of the status.

02047   CZUDC HRD ERR  02047 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
        DISK RESIDENT  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DRIVE CLEAR COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
          EXPECTED RESPONSE  7E
          ACTUAL RESPONSE    00

              The DRIVE CLEAR command did not return an expected
              response code.  The expected response and actual response
              are in hex.
```

3.2.5 TEST 3 INFORMATIONAL MESSAGES

```
        UNIT xx UDA AT xxxxxx DRIVE xxx   RUNTIME hhh:mm:ss
        LOGGABLE INFORMATION AFTER RECAL
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

              After sending a RECALIBRATE command, the ATTENTION
              bit was set.  Test 3 then sent a GET STATUS command
              and found the LOGGABLE INFORMATION bit was set.  This
              is not an error, it is only some information being sent
              from the drive.  Normal operation continues.

              Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'
```

3.2.6 TEST 3 ERROR MESSAGES (03000 TO 03999)

03001 CZUDC HRD ERR 03001 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
TIME-OUT ON SEND
COMMAND WAS command
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

If test 3 tries to send a level 2 command to the drive, and receiver ready is deasserted, error 3001 occurs. Where command is one of the following:

GET COMMON CHARACTERISTICS
ONLINE
DRIVE CLEAR
DISCONNECT
GET SUBUNIT CHARACTERISTICS
GET STATUS
CHANGE MODE
INITIATE RECLIBRATE
SPIN UP

REAL TIME STATE state: REAL TIME STATE 0003
The real time state is the real time drive state <<AFTER>> Test 3 detected the error. <<THIS VALUE IS DISPLAYED IN HEX>>. In this example, receiver ready and attention are both asserted.

The bit positions are defined as follows:
0001 - Receiver ready (Test 3 able to transmit to drive)
0002 - Attention (error occurred or online timeout expired)
0040 - Available (drive offline and usuable)
1000 - Read/Write ready
The complete meaning of these bits is beyond the scope of this text, please refer to the operator documentation for the drive you are working on.

STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0:
The status is the response to the SDI GET STATUS command. These words are printed in HEX. <<NOTE THAT THE STATUS IS PRINTED OUT FROM RIGHT TO LEFT!!>>. The status' meaning is beyond the scope of this text, please refer to the operator documentation for the drive you are working on.

```
03002   CZUDC HRD ERR  03002 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        TIME-OUT OF RECEIVE
        COMMAND WAS GET COMMON CHARACTERISTICS
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                This error is a failure of the drive to respond to an SDI
                level 2 command (see the SDI specification) befor the
                drive-supplied command timeout expires.

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

03003   CZUDC HRD ERR  03003 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        FIRST WORD RECEIVED WAS NOT A START FRAME
        COMMAND WAS GET COMMON CHARACTERISTICS
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The first word received by the UDA from the drive was not a
                valid message start frame.


03004   CZUDC HRD ERR  03004 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        FRAMING ERROR ON LEVEL 0 RESPONSE
        COMMAND WAS GET COMMON CHARACTERISTICS
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                Error 3004 is caused by one or more of the following
                conditions:  1) Illegal frame code -- the frame is not
                a message start, continue, or end frame.  2) Illegal
                sequence of frames -- such as a message start frame
                without ever receiving a message end frame.  This can be
                caused by the drive sending a response before the UDA
                asserts receiver ready, or a random hit on the SDI cable
                that garbles a frame or a bad drive transmitter or UDA
                receiver.

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

03005   CZUDC HRD ERR  03005 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        CHECKSUM ERROR ON LEVEL 0 RESPONSE
        COMMAND WAS GET COMMON CHARACTERISTICS
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The checksum attached to a message end frame did not match
                the checksum computed over the level 2 command.  This could
                be caused by a bad drive transmitter, bad UDA receiver,
                incorrectly computed checksum by the drive (unlikely)
                or a random hit on the SDI cable.

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'
```

```
03006   CZUDC HRD ERR  03006 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        RESPONSE LONGER THAN EXPECTED
        COMMAND WAS GET COMMON CHARACTERISTICS
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The buffer size set aside for the response was not large
                enough for the response received.  This is caused by the
                drive sending a response that is incorrect for the request
                sent to the drive,  or the drive sending some garbage with
                the response.

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

03007   CZUDC HRD ERR  03007 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        CODE FROM RECEIVE WAS UNINELLIGIBLE FROM SUBSYSTEM = 0000
        COMMAND WAS GET COMMON CHARACTERISTICS
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The unknown error code occurs when the UDA returns an error
                code from an operation that test 3 does not recognize.
                Possible UDA microcode change without test 3 update.

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

03008   CZUDC HRD ERR  03008 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
        COMMAND WAS GET COMMON CHARACTERISTICS
          EXPECED RESPONSE    7E
          ACTUAL RESPONSE     7D
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                This is caused by receiving an UNSUCCESSFUL response from
                the drive, or the drive sending some response other than
                the correct response for the request sent to the drive.
                See the contents of status for the unexpected response
                error (or reason).

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

03009   CZUDC HRD ERR  03009 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DRIVE NOT ASSERTING RECEIVER READY IN DRIVE STATE
        REAL TIME STATE 0003

                Test 3 inits the drive and checks the drive's real time
                state.  If RECEIVER READY was not asserted after a period
                of time this error message is printed.
```

03010 CZUDC HRD ERR 03010 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
FAILED TO RECEIVE VALID DRIVE STATE
REAL TIME STATE 0003

There are two types of invalid state: no clocks or 'hard' errors. If after getting state and no clocks occur, error 3010 is reported. Check the drive state for further information.

03011 CZUDC HRD ERR 03011 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
CANNOT RECEIVE DRIVE STATE FROM DRIVE
CHECK IF DRIVE IS POWERED ON.
REAL TIME STATE 0003

After the test 3 sends the drive a DISCONNECT command test 3 should be able to receive state from the drive. The drive may have spun down after the DISCONNECT command.

03012 CZUDC HRD ERR 03012 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
DRIVE STATE RECEIVED HAS BAD PARITY
REAL TIME STATE 0003

As in 3010, we can get two types of invalid state. If parity or pulse errors occur for 1/2 a second, either the transmitter or receiver is bad. This could be caused by a bad transmitter or receiver or by a hit on the SDI cable.

03013 CZUDC DVC FTL 03013 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
NO VALID STATE FROM DRIVE
REAL TIME STATE 0003

The drive recieved either one of the two types of invalid state that are described in 3010 and 3012. Check state for further information. This could be caused by a bad transmitter or receiver or by a hit on the SDI cable.

03014 CZUDC HRD ERR 03014 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
SUBUNIT CHARACTERISICS SAY THERE ARE ZERO READ ONLY GROUPS
IN THE DIAGNOSTIC AREA

After interrogating the subunit characteristics, test 3 finds out that the drive claims there are zero read only groups in the diagnostic area. There must be at least one for the test to run.

03015 CZUDC HRD ERR 03015 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
SUBUNIT CHARACTERISTICS SAY THERE ARE LESS THAN 1 READ/WRITE
GROUPS IN THE DIAGNOSTIC AREA

After interrogating the subunit characteristics, test 3 finds out that the drive claims there are zero read/write groups in the diagnostic area. There must be at least one for the test to run.

03016 CZUDC HRD ERR 03016 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
NEITHER R/W READY NOR ATTENTION SET AFTER RECALIBRATE COMMAND
REAL TIME STATE 0003

After a RECALIBRATE command, R/W READY or ATTENTION did not set. Check the state for further information. This could be cause by a bad transmitter or receiver or by a hit on the SDI cable.

03017 CZUDC HRD ERR 03017 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
SUBUNIT CHARACERISTICS SAY LESS THAN 1 DIAGNOSTIC CYLINDER

After interrogating the subunit characteristics, test 3 finds out that the drive claims there are zero diagnostic cylinders. There must be at least one for the test to run.

03018 CZUDC HRD ERR 03018 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
READ/WRITE READY DROPPED BEFORE FORMAT OPERATION

The R/W READY signal was deasserted by the drive before a format operation was going to be sent by the UDA. The drive may have gone off line or is not transmitting properly or the UDA may not be receiving properly or the SDI cable took a hit.

03019 CZUDC HRD ERR 03019 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
FORMAT OPERATION REPORTED TIME-OUT FAILURE
CYLINDER aaa. GROUP bb. TRACK cc.

The format operation sent by the UDA failed. The command timed out possibly due to receiver ready being dropped or communication problem (bad transmitter or receiver or hit on the SDI cable)

Where:

aaa is the cylinder value in decimal.
bb is the group value in decimal.
cc is the track value in decimal.

```
03020   CZUDC HRD ERR  03020 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        AFTER RECAL, ERROR BITS WERE SET
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                After sending a RECALIBRATE command, the ATTENTION
                bit was set.  Test 3 then sent a GET STATUS command
                and found the error bits were set.  For further
                information, check the state and the status.

                Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

03022   CZUDC HRD ERR  03022 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        READ/WRITE READY DROPPED BEFORE WRITE OPERATION

                The R/W READY signal was deasserted by the drive before
                a write operation was going to be sent by the UDA.
                The drive may have gone off line or is not transmitting
                properly or the UDA may not be receiving properly or
                the SDI cable took a hit.

03023   CZUDC HRD ERR  03023 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK.  ON LAST BLOCK:
        WRITE OPERATION REPORTED FAILURE -- ERROR CODE aaa OCTAL.
        DBN bbb.  CYLINDER ccc.  GROUP dd.  TRACK ee.

                After each track in the diagnostic space is formatted, at
                least one block must be able to have data written to it
                and read from it and the data must be correct. Not one
                block (DBN bbb.) from track (ee) was able to pass.  The
                error code (aaa) gives the reason for the write operation
                failure.

                Where:
                        aaa is the error code in octal.
                                It may have one of the following values:
                                        2 = drive failure
                                        3 = requested LBN is a secondary
                                            revector.
                                 <<< NOTE >>> We are working with DBN's
                                        4 = header compare failure
                                            (desired header not found)
                                        153 = suspected positioner error
                                        213 = read/write ready failure
                                        253 = drive data or state clock timeout
                                            (indicates cable/transmitter/
                                             receiver broken)
                                        313 = receiver ready timeout
                                        413 = drive state receive error during write
                        bbb is the DBN in decimal.
                        ccc is the cylinder value in decimal.
                        dd is the group value in decimal.
                        ee is the track value in decimal.
```

03024 CZUDC HRD ERR 03024 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
READ/WRITE READY DROPPED BEFORE READ OPERATION

The R/W READY signal was deasserted by the drive before a read operation was going to be sent by the UDA. The drive may have gone off line or is not transmitting properly or the UDA may not be receiving properly or the SDI cable took a hit.

03025 CZUDC HRD ERR 03025 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK. ON LAST BLOCK:
READ OPERATION REPORTED FAILURE -- ERROR CODE aaa OCTAL.
DBN bbb. CYLINDER ccc. GROUP dd. TRACK ee.

After each track in the diagnostic space is formatted, at least one block must be able to have data written to it and read from it and the data must be correct. Not one block (DBN bbb.) from track (ee) was able to pass. The error code (aaa) gives the reason for the read operation failure.

Where:

aaa is the error code in octal.
It may have one of the following values:
2 = drive failure
3 = requested LBN is a secondary revector.
<<< NOTE >>> We are working with DBN's
4 = header compare failure (desired header not found)
52 = SERDES overrun error
150 = data sync timeout on read
153 = suspected positioner error
213 = read/write ready failure
253 = drive data or state clock timeout (indicates cable/transmitter/receiver broken)
313 = receiver ready timeout
413 = drive state receive error during write

bbb is the DBN in decimal.
ccc is the cylinder value in decimal.
dd is the group value in decimal.
ee is the track value in decimal.

```
03026   CZUDC HRD ERR  03026 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK. ON LAST BLOCK:
        DATA COMPARE FAILURE ON WORD aa.
        EXPECTED DATA  bbbb
        ACTUAL DATA    cccc
        CYLINDER ddd.  GROUP ee.  TRACK ff.

                After each track in the diagnostic space is formatted, at
                least one block must be able to have data written to it
                and read from it and the data must be correct. Not one
                block (DBN bbb.) from track (ee) was able to pass.  The
                data read did not match the data written.

                Where:
                        aa is the offset in decimal into the buffer where
                                the error occurred.
                        bbbb is the expected data in hex.
                        cccc is the actual data in hex.
                        ddd is the cylinder value in decimal.
                        ee is the group value in decimal.
                        ff is the track value in decimal.


03027   CZUDC HRD ERR  03027 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        SEEK COMPLETE TIME-OUT -- READ/WRITE READY DID NOT SET
        SEEK WAS TO CYLINDER aaa.  GROUP bb.

                After a SEEK command has been successfully sent from the UDA
                to the drive, the signal READ/WRITE READY must be set to
                indicate that the seek completed.  If READ/WRITE READY
                never is asserted by the drive after the seek, the seek
                times out and error 3027 is presented.

                Where:
                        aaa is the cylinder in decimal.
                        bb is the group in decimal.

03028   CZUDC HRD ERR  03028 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        NO BLOCK ON THIS TRACK CAN BE READ. LAST BLOCK TRIED:
        aBN bbbb.  CYLINDER ccc.  GROUP dd.  TRACK ee.

                After a seek to a track, at least one block must be able
                to be read to assure that test 3 can read the header. If
                not one block was successful, error message 3028 appears.

                Where:
                        a is 'L' for LBN, 'D' for DBN, or 'X' for XBN.
                        bbbb is the block number in decimal.
                        ccc is the cylinder in decimal.
                        dd is the group number in decimal.
                        ee is the track number in decimal.
```

03029 CZUDC HRD ERR 03029 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
AVAILABLE WAS NOT ASSERTED AFTER DISCONNECT
STATE RECEIVED state

After the DISCONNECT command was sent, the AVAILABLE flag should be asserted after a period of time. It it never was, then error 3029 appears. There maybe a problem with a transmitter or a receiver or the SDI cable at this point.

03030 CZUDC HRD ERR 03030 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
INVALID COMMAND aaaa WAS SUCCESSFUL

Some invalid level 2 commands are sent over the SDI. The drive should find these illegal commands and flag them as such. If the drive doesn't, then error 3030 will appear.

Where aaaa is the invalid command in hex.

03031 CZUDC HRD ERR 03031 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
COMMAND WITH type LENGTH = a WAS SUCCESSFUL

SDI level 2 commands with invalid lengths are sent to the drive to check if the drive can find them.

Where:
type could be 'COMMAND' or 'RESPONSE' for which field was affected
a is the invalid length

03032 CZUDC HRD ERR 03032 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
UNIT DID NOT REPORT TRANSMITTION ERROR
WHEN reason

Invalid level 1 sequences were sent to the drive. Several sequences are tried and the drive should find fault with everyone of them.

Where reason could be one of the following:

AN END FRAME WAS SENT AFTER A START FRAME TIMED OUT
A CONTINUE OR END FRAME DID NOT FOLLOW A START FRAME
AN END FRAME WAS SENT WITH NO START FRAME
AN END FRAME WIH A BAD CHECKSUM WAS SENT
A CONTINUE FRAME WAS SENT WITH NO START FRAME

03033 CZUDC HRD ERR 03033 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
DISK FUNCTION DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
UNIT ACCEPTED AN INVALID GROUP NUMBER FROM GROUP SELECT LEVEL 1

A level 1 select group command with an illegal group number is sent to the drive. If the drive accepted it, then error 3033 will be displayed.

```
03034   CZUDC DVC FTL ERR  03034 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        UNABLE TO CORRECTLY READ OVERLAY x
        THIS UDA AND ALL DRIVES ATTACHED WILL BE REMOVED FROM TESTING

                There are two overlays in test 3.  For some reason that
                the overlay cannot be read correctly, error 3034 will be
                displayed.  Since no code can be loaded into the UDA at
                this point, the UDA and all attached drives will cease to
                be tested.  The reason for this may be bad UNIBUS memory
                or board 1 may be failing.

                <<< NOTE >>>    This is -- NOT -- a drive failure.

03035   CZUDC DVC FTL ERR  03035 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        SUCCESSFULLY WROTE ON DBN AREA WHEN DRIVE WAS WRITE PROTECTED

                An attempt was made to write on a write protected
                drive. It should have resulted in an error response
                from the disk drive, but it didn't.

03036   CZUDC DVC FTL ERR  03036 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DRIVE IS NOT PROPERLY FORMATTED.
        UDA WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION
        THIS DRIVE NEEDS TO BE FORMATTED.

                Test 3 reads a copy of the FCT in the XBN area and determined
                that the FCT was corrupted.  Any normal operating system
                (which uses the UDA as a controller) will spin down the drive,
                so the drive will need to be reformatted.

03037   CZUDC DVC FTL ERR  03037 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        DRIVE IS FORMATTED IN 576 BYTE MODE.
        TO RUN WITH A UDA, THIS DRIVE NEEDS TO BE FORMATTED
        IN 512 BYTE MODE.

                Test 3 read a copy of the FCT from the XBN area and determined
                that the drive was formatted in 576 byte mode.  Any normal
                operating system (which uses the UDA as a controller) will spin
                down the drive, so the drive will need to be reformatted.

03038   CZUDC DVC FTL ERR  03038 ON UNIT 00 TST 003 SUB 000 PC: xxxxxx
        DISK FUNCTION DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hhh:mm:ss
        NO COPY OF THE FCT COULD BE READ.
        UDA WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION
        THIS DRIVE NEEDS TO BE FORMATTED.

                Test 3 attempted to read every copy of the FCT without success.
                Any normal operating system (which uses the UDA as a controller)
                will spin down the drive, so the drive will need to be reformatted
```

3.2.7 TEST 4 INFORMATIONAL MESSAGES

```
UNIT u UDA AT cccccc DRIVE n   RUNTIME hh:mm:ss
A CORRECTABLE ECC ERROR EXISTS IN type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
```

The above message occurs when Test 4 1) detects an ECC error and 2) is able to correct it, and 3) the corrections are less than the drive ECC threshold, (a SDI DRIVE CHARACTERISTIC) and 4) the EDC computed over the corrected sector matched the EDC read.

```
UNIT unit UDA AT udaadr DRIVE plug   RUNTIME hh:mm:ss
INITIAL WRITE COMPLETE
```

Whenever Test 4 is STArted with initial write enabled, <<OR>> whenever it is STArted or REStarted and the diagnostic area is being tested or a drive not in read only mode, the disk will be initially written. The above message occurs when the initial write completes.

```
UNIT unit UDA AT udaadr DRIVE plug   RUNTIME hh:mm:ss
READ ONLY DRIVE, INITIAL WRITE WILL NOT BE PERFORMED
```

If an initial write is to be performed (see above for conditions) and a unit or subunit is in read only mode, (can be set in the manual intervention questions) an initial write will not be performed, and this message will print to inform the operator.

NOTE: DATA COMPARE ERRORS RESULT IF THE DISK IS NOT INITIALLY WRITTEN!!

```
UNIT unit UDA AT udaadr DRIVE plug   RUNTIME hh:mm:ss
THE PREVIOUS DEVICE FATAL WILL CAUSE THE FOLLOWING DRIVES
TO BE DROPPED: plug,  plug+1,  plug+2,  plug+3
```

plug: drive plug number -- each subunit's plug number is displayed. for a single subunit drive (such as and RA80) only one plug number is displayed.

If a device fatal error occurs and dropping is enabled, <<ALL>> subunits on the unit that the device fatal occurred must be dropped. To inform the operator, this message is printed after the device fatal error message.

NOTE: IF MORE THAN ONE UDA IS ON A SYSTEM, THIS MESSAGE MAY NOT IMMEDIATELY FOLLOW THE DEVICE FATAL IF AN ERROR HAPPENS AT THE SAME TIME ON ANOTHER UDA.

3.2.8 TEST 4 ERROR MESSAGES (04000 TO 04999)

04001 CZUDC SFT ERR 04001 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
ATTN ASSERTED DURING SEEK -- ERROR OR LOGGABLE INFORMATION
SEEK FROM GRP group CYL cylinder TO GRP group CYL cylinder
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

This error occurs when the drive asserts the SDI ATTENTION signal without asserting the READ/WRITE READY signal, indicating the unsuccessful completion of a seek.

See retry/recovery section for recovery details.

04002 CZUDC SFT ERR 04002 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
ATTN ASSERTED UNEXPECTEDLY, ASYN DRIVE ERROR OR LOGGABLE
INFORMATION -- THIS IS AN <<UNCOUNTED>> SOFT ERROR
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

This is an asynchronous drive error. Asynchronous drive errors are those errors reported by the drive which are not related to a level 2 command. These errors are reported by the drive using the SDI ATTENTION signal. The operator must look at the status returned to determine the error that occurred.

See retry/recovery section for recovery details.

04003 CZUDC SFT ERR 04003 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
SEEK DID NOT COMPLETE, NEITHER ATTN OR R/W RDY WAS ASSERTED
BEFORE TIMEOUT
SEEK FROM GRP group CYL cylinder TO GRP group CYL cylinder
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

This error occurs when the drive fails to assert READ/WRITE READY before the seek timeout, which indicates the successful completion of a seek.

See retry/recovery section for recovery details.

```
04004   CZUDC   HRD ERR  04004 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        RCT AREA CORRUPTED, COULD NOT FIND REPLACEMENT FOR
        LBN THAT WAS REVECTORED
        ATTEMPTING TO READ RCT LBN bn
        SEARCHING FOR LBN bn

        CZUDC   HRD ERR  04004 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        RCT AREA CORRUPTED, COULD NOT FIND REPLACEMENT FOR
        LBN WITH HEADER NOT FOUND
        ATTEMPTING TO READ RCT LBN bn
        SEARCHING FOR LBN bn
```

Error 4004 will occurr only when Test 4 is running in the customer data area. It occurs when 1) A sector is either marked revectored or the header can't be found in two revolutions of the disk (both cases should be revectored) and 2) The replacement for that sector isn't found in the RCT and 3) a NULL entry isn't found at the end of the RCT (see DEC STANDARD 166, Replacement and Caching Table Format). In either case, the subunit should be reformatted, and the cause of the RCT corruption determined.

```
04005   CZUDC   HRD ERR  04005 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        HEADER NOT FOUND DURING WRITE
        DBN bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
```

Error 4005 occurs only when Test 4 is writing a DBN or RBN. This is because bad blocks in the diagnostic area are not revectored, and RBN's are what LBN's are revectord to, so they should never be bad. Test 4 reports this error if the header being searched for couldn't be found in two revolutions of the disk.

```
04006   CZUDC   SFT ERR  04006 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        SELECT TRACK AND WRITE LEVEL 1 CMD NOT EXECUTED
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                Select track and read or write not executed occurs when the
                UDA attempts to send the select track and read/write level 1 cmd,
                but receiver ready is deasserted or the state is invalid so it
                cannot send the command (the SERDES could also be broken so it's unable
                to send the command).  The same error is generated if the UDA gets a
                header sync timeout, and when it looks at the drive's state, it is
                either invalid or reciever ready is deasserted (header sync timeout is
                <<NOT>> a error -- it's quite normal on a high-density disk).

                See retry/recovery section for recovery details.


04007   CZUDC   SFT ERR  04007 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        ECC DETECTED ERROR
        RETRY retry
        ERROR RECOVERY LEVEL level
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder

                Error 4007 occurs if an ECC error is detected but ECC correction is
                disabled.

                See retry/recovery section for recovery details.


04008   CZUDC   SFT ERR  04008 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        ECC DETECTED ERROR, BUT CORRECTION FAILED
        RETRY retry
        ERROR RECOVERY LEVEL level
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder

                Error 4008 occurs if an ECC error is detected, but the correction
                algorithm is unable to correct the errors.

                NOTE: THIS IS USUALLY (BUT NOT ALWAYS) INDICATIVE OF A BAD SPOT IN THE
                ECC RESIDUE AREA AFTER THE DATA AREA OF THE SECTOR.

                See retry/recovery section for recovery details.
```

```
04009   CZUDC   SFT ERR  04009 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        ECC CORRECTIONS EXCEED THRESHOLD
        RETRY retry
        ERROR RECOVERY LEVEL level
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder

                Error 4009 occurs if an ECC error is detected, the correction
                algorithm succeeds in correcting the errors, but the number of bits
                that were corrected exceeds the correction threshold (a SDI DRIVE
                CHARACTERISTIC).

                See retry/recovery section for recovery details.


04010   CZUDC   SFT ERR  04010 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        ECC CORRECTION SUCCEEDED, BUT EDC DETECTS ERROR
        RETRY retry
        ERROR RECOVERY LEVEL level
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder

                Error 4010 could be caused by several problems:

                1) A buffer with a few ECC errors that can be corrected, but the EDC
                was incorrectly computed or written, or 2) The ECC algorithm
                incorrectly corrected the buffer and/or the EDC value, (but corrections
                were less than the threshold) or 3) UDA buffer RAM problem.

                See retry/recovery section for recovery details.


04011   CZUDC   HRD ERR  04011 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        ERROR RECOVERY TRIED ALL LEVELS WITHOUT SUCCESS
        type bn
        GRP group CYL cylinder

                Error 4011 occurs when retries are enabled, and Test 4 has tried
                all retries on all levels of error recovery.  See ECC and EDC
                retries in the retry/recovery section.
```

```
04012   CZUDC   HRD ERR  04012 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        DATA COMPARISON FAILED
        ECC OR EDC HAD DETECTED ERROR IN BUFFER
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        PATTERN NUMBER pattern
        OFFSET OF ERROR WITHIN BUFFER: buffer_offset
        OFFSET OF ERROR WITHIN DISPLAYED LIST: list_offset (1ST WORD OFFSET 0)
            data0   data1   data2   data3   data4   data5
            data6   data7   data8   data9   data10  data11

        CZUDC   HRD ERR  04012 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        DATA COMPARISON FAILED
        ECC OR EDC HAD <<NOT>> DETECTED ERROR IN BUFFER
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        PATTERN NUMBER pattern
        OFFSET OF ERROR WITHIN BUFFER: buffer_offset
        OFFSET OF ERROR WITHIN DISPLAYED LIST: list_offset (1ST WORD OFFSET 0)
            data0   data1   data2   data3   data4   data5
            data6   data7   data8   data9   data10  data11

                pattern:        The pattern number (decimal) that failed the comparison.

                buffer_offset:  The offset of the error (decimal) within the sector read,
                                where the first word in the sector is offset 0

                list_offset:    The offset of the error (decimal) within the displayed list,
                                where the first word in the list is offset 0

                dataX:          Test 4 displays twelve data words read from the sector.
                                They are displayed left to right, top to bottom.

        Error 4012 occurs when a data compare detects a difference between
        the buffer read and a known data pattern.  The operator is informed if
        the error was detected by the ECC or EDC.  The first word of the sector
        which may or may not be printed, depending on the position of the error,
        is the pattern number replicated in each nibble of the word.  If a disk
        is not initally written, it is likely that data comparison failures will
        occur in the fist word of the sector.  The following is the first word
        of the sector for the sixteen different patterns.

              pattern         word 0          pattern         word 0

                1             010421            9             114631
                2             021042           10             125252
                3             031463           11             135673
                4             042104           12             146314
                5             052525           13             156735
                6             063146           14             167356
                7             073567           15             177777
                8             104210           16             000000

        Note that pattern 16 is mapped to pattern 0.
```

04013 CZUDC DEV FTL ERR 04013 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
DRIVE NOT ONLINE TO UDA, AND NOT SPINABLE

If a drive dropps offline while being tested (a normal occurance during Test 4) and some event happens that makes the drive unspinnable (such as the operator popping out the run/stop switch) error 4013 will be printed. If the operator inhibits dropping units, Test 4 will go into error recovery and loop on error 4023, spindle dropped ready.

04014 CZUDC DEV FTL ERR 04014 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
UNABLE TO COMPLETE SEEK -- TRIED 3 TIMES
type bn
GRP group CYL cylinder

Once a seek has been attempted 3 times, and never successfully completed, error 4014 will be printed and the entire unit dropped. If the operator inhibits dropping units, the drive will be recalibrated, and the seek will be attempted again.

04015 CZUDC SFT ERR 04015 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
SEEK REQUIRED retries RETRIES BEFORE COMPLETING
GRP group CYL cylinder

retries: The number of times the seek was re-issued

If a seek required retries, error 4015 would print to notify the operator.

04016 CZUDC DEV FTL ERR 04016 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
ERRORS DURING DRIVE INITIALIZATION AND SETUP
THIS UDA AND ALL DRIVES ATTACHED WILL BE REMOVED FROM TESTING

If any errors occur during drive and test initialization, DRIVES ATTACHED TO THE UDA THAT HAD THE DRIVE INITIALIZATION ERRORS WILL NOT BE TESTED. In this case, error 4016 will be printed to notify the operator. THIS ERROR DOES <<NOT>> REFER TO UDA INITIALIZATION. This error is unaffected by the operator inhibiting the dropping of units.

```
04017   CZUDC   DEV FTL ERR  04017 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        NO VALID STATE FROM DRIVE
        NO DRIVE CLOCKS

        CZUDC   DEV FTL ERR  04017 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        NO VALID STATE FROM DRIVE
        HARD PARITY OR PULSE ERROR FOR 1/2 A SECOND
```

If Test 4 is <<EVER>> unable to get valid drive state, the drive is immediately dropped, and error 4017 is printed. There are two types of invalid state: no clocks or 'hard' errors. If Test 4 <<EVER>> detects no clocks, the driver is dropped IMMEDIATELY. Parity and pulse errors are normal, so Test 4 tolerates them, <<UNLESS THEY HAPPEN CONTINUOUSLY FOR 1/2 A SECOND>>. If they do occur for 1/2 a second, either the transmitter or receiver is bad, and the drive is dropped. If the operator has inhibited the dropping of units, Test 4 will retry the module that the error occurred on.

```
04018   CZUDC   DEV FTL ERR  04018 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        ATTEMPT TO WRITE ON WRITE PROTECTED DRIVE
        ERROR CODE RETURNED FROM UDA: code
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

code: The error (in octal) returned to Test 4 from the UDA when Test 4 attempted to write on the write protected drive.

The UDA error codes (in octal) are as follows:

code	error
2	SELECT TRACK AND WRITE LEVEL 1 CMD NOT EXECUTED
3	LBN IS REVECTORED
4	HEADER NOT FOUND
153	SEEK OR HEAD SELECT ERROR
213	R/W RDY DROPPED
253	DATA OR STATE CLOCK TIMEOUT
313	RCVR RDY DROPPED
413	REAL TIME STATE RECEIVE ERROR

If an attempt is made to write on a write protected drive, the drive <<SHOULD>> drop READ/WRITE READY -- this is an error code of 213. If <<ANY>> other code is returned from the drive, the drive is causing the write to fail in an incorrect manner.

If Test 4 attempts to write on a write protected drive, error 4018 is printed. Test 4 requires the drive to detect the attempt to write when write protected and return an error for this error to be printed. If the operator has inhibited the dropping of units, a seek will be issued and the write attempted again.

```
04019   CZUDC   HRD ERR  04019 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        HEADER NOT FOUND DURING READ
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
```

Error 4019 occurs only when Test 4 is reading a DBN or RBN. This is because bad blocks in the diagnostic area are not revectored, and RBN's are what LBN's are revectord to, so they should never be bad. Test 4 reports this error if the header being searched for couldn't be found in two revolutions of the disk.

```
04020   CZUDC   SFT ERR  04020 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        SELECT TRACK AND READ LEVEL 1 CMD NOT EXECUTED
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

Select track and read or write not executed occurs when the UDA attempts to send the select track and read/write level 1 cmd, but receiver ready is deasserted or the state is invalid so it cannot send the command (the SERDES could also be broken so it's unable to send the command). The same error is generated if the UDA gets a header sync timeout, and when it looks at the drive's state, it is either invalid or reciever ready is deasserted (header sync timeout is <<NOT>> a error -- it's quite normal on a high-density disk).

See retry/recovery section for recovery details.

```
04021   CZUDC   DEV FTL ERR  04021 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        DRIVE NOT FORMATTED IN 512 BYTE MODE -- UNABLE TO TEST
        XBN 0 MODE WORD: mode
```

mode: The mode word found on the drive's XBN 0

Error 4021 occurs only when Test 4 is going to test in the customer data area, and the mode word found in XBN 0 is not the 512 byte mode word (126736 octal). See DEC STANDARD 166 "FCT Structure". Inhibiting the dropping of units has no effect on this error.

```
04023   CZUDC   DEV FTL ERR  04023 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        UNABLE TO CONTINUE TESTING
        PORT SWITCH OUT
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

If, during testing, the operator disables the port that Test 4 is using by popping out the port switch, Test 4 prints error 4023. CHANGING THE STATE OF THE PORT SWITCH FOR THE PORT THAT Test 4 IS <<NOT>> USING HAS NO EFFECT ON THE TEST. If dropping of units is inhibited, Test 4 will loop in error recovery, printing this error, until the error state is corrected (by some external action).

```
        CZUDC   DEV FTL ERR  04023 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        UNABLE TO CONTINUE TESTING
        RUN/STOP SWITCH OUT
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

If, during testing, the operator pops out the run/stop switch, Test 4 prints error 4023. If dropping of units is inhibited, Test 4 will loop in error recovery, printing this error, until the error state is corrected (by some external action).

```
        CZUDC   DEV FTL ERR  04023 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        UNABLE TO CONTINUE TESTING
        SPINDLE DROPPED READY
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

If, during testing, the spindle drops from its ready state, error 4023 is printed. If dropping of units is inhibited, Test 4 will loop in error recovery, printing this error, until the error state is corrected (by some external action).

```
04024   CZUDC   SFT ERR  04024 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        EDC DETECTED ERROR BUT ECC DID NOT
        RETRY retry
        ERROR RECOVERY LEVEL level
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        EDC COMPUTED edc EDC READ edc

                edc:            The edc computed and read in octal.

                Error 4024 could be caused by several problems.  1) A buffer with no
                ECC errors, but the EDC was incorrectly computed or written, or 2) UDA
                buffer RAM problem, or 3) The error is such that the ECC really doesn't
                detect an error... This is unlikely.

                See retry/recovery section for recovery details.


04025   CZUDC   HRD ERR  04025 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        WRITE ATTEMPTED MAXIMUM TIMES
        type bn

                If three I/O errors occur when attempting to write to the drive
                (one I/O error if retries are disabled) error 4025 is printed
                to inform the operator.


04026   CZUDC   HRD ERR  04026 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        READ ATTEMPTED MAXIMUM TIMES
        type bn

                If three I/O errors occur when attempting to read from the drive
                (one I/O error if retries are disabled) error 4026 is printed
                to inform the operator.


04028   CZUDC   DEV FTL ERR  04028 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        BOTH READ ONLY <AND> WRITE ONLY BITS SET -- HOST ERROR

                Error 4028 prints ONLY IF THERE IS A HOST CODE ERROR -- THIS IS NOT
                AN ERROR FROM A DRIVE.  Inhibiting the dropping of units has no effect
                on this error.
```

04033 CZUDC DEV FTL ERR 04033 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
UNABLE TO CORRECTLY READ OVERLAY overlay_number
THIS UDA AND ALL DRIVES ATTACHED WILL BE REMOVED FROM TESTING

overlay_number: The overlay number in octal that could not be read.

Because of Test 4's size, most of the program is stored in host memory and is overlay driven. If any error is detected during a UNIBUS read of an overlay, Test 4 will retry the read (with no error report). It will attempt to read an overlay three times before error 4033 is printed, and the test immediately halted. This error can have several causes: 1) the UNIBUS died (it's improbable that you even get the message in this case) or 2) the UDA's UNIBUS interface died (also unlikely that you get a message), or 3) the host program wiped out the Test 4 overlays (since they are stored in host memory - most likely) or 4) a host memory problem - also likely. Inhibiting the dropping of units has no effect on this error.

04034 CZUDC SFT ERR 04034 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
SERDES OVERRUN ERROR DURING READ
ATTEMPT attempt
type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
ORIGIN OF SEEK: GRP group CYL cylinder
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The SERDES overrun error is detected on a read operation and is indicative of a drive whose transfer rate is greater than 23 MHZ or a broken SERDES.

See retry/recovery section for recovery details.

04035 CZUDC SFT ERR 04035 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
DATA OR STATE CLOCK TIMEOUT DURING READ
ATTEMPT attempt
type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
ORIGIN OF SEEK: GRP group CYL cylinder
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The loss of drive clock occurs when the UDA is clocking data to or from the drive through the SERDES. Failure of a word to be clocked in during a 125 millisecond time period triggers a loss of drive clock error.

See retry/recovery section for recovery details.

```
04036   CZUDC   SFT ERR  04036 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        DATA SYNC TIMEOUT DURING READ
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                This error occurs on a read operation after the correct header has
                been found and the UDA times out waiting for the data sync word.

                See retry/recovery section for recovery details.


04037   CZUDC   SFT ERR  04037 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        R/W RDY DROPPED DURING READ
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The loss of read/write ready error is detected either before an
                I/O has begun when trying to send out the real time command or at
                the end of an I/O operation when checking for errors.

                See retry/recovery section for recovery details.


04038   CZUDC   SFT ERR  04038 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        RCVR RDY DROPPED DURING READ
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The loss of drive receiver ready is detected when the UDA is trying
                to send out a real-time read or write command.

                See retry/recovery section for recovery details.
```

04040 CZUDC HRD ERR 04040 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
ALL COPIES OF RCT READ WITH ERROR, SEARCHING FOR
LBN THAT WAS REVECTORED
LAST RCT LBN SEARCHED bn
SEARCHING FOR LBN bn

CZUDC HRD ERR 04040 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
ALL COPIES OF RCT READ WITH ERROR, SEARCHING FOR
LBN WITH HEADER NOT FOUND
LAST RCT LBN SEARCHED bn
SEARCHING FOR LBN bn

Error 4040 occurs when Test 4 is trying to find the RBN that replaces a LBN that was revectored or whose header could not be found (both should be revectored). Test 4 was unable to get a valid copy out of the M copies of the RCT due to I/O errors or ECC/EDC errors. M is a SDI DRIVE CHARACTERISTIC and is defined by the drive. This is indicitave of either a bad pack (HDA) or that something wrote over the RCT incorrectly. Try to reformat the subunit.

04041 CZUDC HRD ERR 04041 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
COULD NOT FIND REPLACEMENT FOR
LBN THAT WAS REVECTORED
LBN TO REPLACE bn

CZUDC HRD ERR 04041 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
COULD NOT FIND REPLACEMENT FOR
LBN WITH HEADER NOT FOUND
LBN TO REPLACE bn

Error 4041 only occurs when Test 4 is running in the customer data area, and is trying to find the RBN that replaces a LBN that was revectored (must be in the RCT) or whose header could not be found (should be in the RCT, unless the media under the header has 'grown' a bad spot recently). In either case, Test 4 was unable to find an entry in the RCT for the the sector and the subunit should be reformatted. In the case of the revectored LBN, the cause of the RCT's corruption should be determined (even with the header not found, the RCT may have been corrupted because a header going bad without warning [eg. the formatter not being able to see it as a weak spot] is a very low probibility occurance).

```
04042   CZUDC   DEV FTL ERR  04042 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        TIMEOUT WAITING FOR SECTOR OR INDEX PULSE
        GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                Error 4042 occurs when the UDA microcode never detects a sector
                or index pulse from the drive before a read or write operation.
                If dropping of units is inhibited, a seek will be issued, and
                the write attempted again.

04044   CZUDC   SFT ERR  04044 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        SEEK OR HEAD SELECT ERROR DETECTED DURING WRITE
        ATTEMPT attempt
        LBN bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                See error 4045 for description.

                See retry/recovery section for recovery details.

04045   CZUDC   SFT ERR  04045 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        SEEK OR HEAD SELECT ERROR DETECTED DURING READ
        ATTEMPT attempt
        LBN bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                Errors 4044 and 4045 occur when the header comparison routine determines
                that the drive is positioned at the wrong physical cylinder, or that the
                wrong head (which can be cylinders, groups or tracks, or any combination
                depending on the drive) had been selected.  This error only occurs when
                the drive itself had not detected the misseek or incorrect head selected.

                NOTE:   These errors will only be detected when the operator is running
                        Test 4 in the customer data area.  This error will <<never>> appear
                        when running in the diagnostic area.

                See retry/recovery section for recovery details.
```

```
04047   CZUDC   SFT ERR  04047 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        DATA OR STATE CLOCK TIMEOUT DURING WRITE
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The loss of drive clock occurs when the UDA is clocking data to or
                from the drive through the SERDES.  Failure of a word to be
                clocked in during a 125 millisecond time period triggers a loss of
                drive clock error.

                See retry/recovery section for recovery details.


04048   CZUDC   SFT ERR  04048 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        R/W RDY DROPPED DURING WRITE
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The loss of read/write ready error is detected either before an
                I/O has begun when trying to send out the real time command or at
                the end of an I/O operation when checking for errors.

                See retry/recovery section for recovery details.


04049   CZUDC   SFT ERR  04049 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        RCVR RDY DROPPED DURING WRITE
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                The loss of drive receiver ready is detected when the UDA is trying
                to send out a real-time read or write command.

                See retry/recovery section for recovery details.
```

```
04050   CZUDC   DEV FTL ERR  04050 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
        BEGIN/END SET STARTING BLOCK NUMBER GREATER THAN ENDING BLOCK NUMBER

                This is a Test 4 initialization error due to an operator error.  Go back
                to the manual intervention questions and check the answers to the
                BEGIN/END set questions.  Inhibiting the dropping of units has no effect
                on this error.


04051   CZUDC   DEV FTL ERR  04051 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
        THE BEGIN/END SETS OVERLAP

                This is a Test 4 initialization error due to an operator error.  Go back
                to the manual intervention questions and check the answers to the
                BEGIN/END set questions.  Inhibiting the dropping of units has no effect
                on this error.


04052   CZUDC   DEV FTL ERR  04052 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
        BEGIN/END SET ENDING BLOCK NUMBER EXCEEDS MAXIMUM
        MAXIMUM BLOCK NUMBER ON DEVICE IS maximum_block_number

                maximum_block_number:   This is the highest block number the operator
                                        can specify.

                This is a Test 4 initialization error due to an operator error.  Go back
                to the manual intervention questions and check the answers to the
                BEGIN/END set questions.  Inhibiting the dropping of units has no effect
                on this error.


04053   CZUDC   DEV FTL ERR  04053 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
        DUPLICATE BAD BLOCKS

                This is a Test 4 initialization error due to an operator error.  Go back
                to the manual intervention questions and check the answers to the
                BAD BLOCK questions.  Inhibiting the dropping of units has no effect
                on this error.
```

04054 CZUDC DEV FTL ERR 04054 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
BAD BLOCK NUMBER EXCEEDS MAXIMUM. MAXIMUM BLOCK NUMBER
ON DEVICE IS maximum_block_number

maximum_block_number: This is the highest block number the operator can specify.

This is a Test 4 initialization error due to an operator error. Go back to the manual intervention questions and check the answers to the BAD BLOCK questions. Inhibiting the dropping of units has no effect on this error.

04055 CZUDC DEV FTL ERR 04055 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
STARTING CYLINDER GREATER THAN ENDING CYLINDER

This is a Test 4 initialization error due to an operator error. Go back to the manual intervention questions and check the answers to the STARTING AND ENDING CYLINDER questions. Inhibiting the dropping of units has no effect on this error.

04056 CZUDC DEV FTL ERR 04056 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
RANDOM AND SEQUENTIAL SEEKS CANNOT BE MIXED WITHIN A UNIT

Error 4056 is an operator error. The error occurs on a multiple subunit drive when one subunit is selected to run in random mode, and another is selected to run in sequential mode. This mix is not supported, so the above message is issued. Inhibiting the dropping of units has no effect on this error.

04057 CZUDC DEV FTL ERR 04057 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
OVERFLOW WHEN CALCULATING THE L/DBN FROM THE GIVEN CYLINDER
CYLINDER TOO LARGE

This is a Test 4 initialization error due to an operator error. The operator entered a cylinder number, that when converted to a block number, the block number exceeded (2**28) - 1. Go back to the manual intervention questions and check the answers to the STARTING AND ENDING CYLINDER questions. Inhibiting the dropping of units has no effect on this error.

04058 CZUDC DEV FTL ERR 04058 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
TRACK EXCEEDS MAXIMUM FOR DEVICE. MAXIMUM IS maximum_track

maximum_track: This is the highest track number the operator can specify.

This is a Test 4 initialization error due to an operator error. Go back to the manual intervention questions and check the answers to the TRACK questions. Inhibiting the dropping of units has no effect on this error.

CZUDC DEV FTL ERR 04058 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
GROUP EXCEEDS MAXIMUM FOR DEVICE. MAXIMUM IS maximum_group

maximum_group: This is the highest group number the operator can specify.

This is a Test 4 initialization error due to an operator error. Go back to the manual intervention questions and check the answers to the GROUP questions. Inhibiting the dropping of units has no effect on this error.

04059 CZUDC DEV FTL ERR 04059 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
TWO IDENTICAL TRACKS

This is a Test 4 initialization error due to an operator error. Go back to the manual intervention questions and check the answers to the TRACK questions. Inhibiting the dropping of units has no effect on this error.

CZUDC DEV FTL ERR 04059 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
TWO IDENTICAL GROUPS

This is a Test 4 initialization error due to an operator error. Go back to the manual intervention questions and check the answers to the GROUP questions. Inhibiting the dropping of units has no effect on this error.

04062 CZUDC DEV FTL ERR 04062 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
DBN COMPUTED FROM END CYLINDER GIVEN EXCEEDS MAXIMUM DBN NUMBER ON
DEVICE - CYLINDER TOO LARGE

This is a Test 4 initialization error.
Note that though there may be writeable DBN's on the 'last' cylinder, the read only diagnostic area may start on that same cylinder, and Test 4 tries to write to the end of the cylinder that the operator specified. Therefore, specify the previous cylinder if cylinders must be specified. Inhibiting the dropping of units has no effect on this error.

CZUDC DEV FTL ERR 04062 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
OPERATOR ERROR IN ANSWERING MANUAL INTERVENTION QUESTIONS FOR THIS UNIT
LBN COMPUTED FROM END CYLINDER GIVEN EXCEEDS MAXIMUM LBN NUMBER ON
DEVICE - CYLINDER TOO LARGE

This is a Test 4 initialization error.
Note that though there may be writeable LBN's on the 'last' cylinder, the RCT area may start on that same cylinder, and Test 4 tries to write to the end of the cylinder that the operator specified. Therefore, specify the previous cylinder if cylinders must be specified. Inhibiting the dropping of units has no effect on this error.

04063 CZUDC SFT ERR 04063 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
REAL TIME STATE RECEIVE ERROR DURING WRITE
ATTEMPT attempt
type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
ORIGIN OF SEEK: GRP group CYL cylinder
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

The real time drive state receive error is detected at the end of an I/O operation and indicates that there was a pulse or parity error in the receipt of the drive's state during the I/O operation.

See retry/recovery section for recovery details.

```
04064   CZUDC   SFT ERR  04064 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        REAL TIME STATE RECEIVE ERROR DURING READ
        ATTEMPT attempt
        type bn
        SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
        ORIGIN OF SEEK: GRP group CYL cylinder
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

The real time drive state receive error is detected at the end of an I/O operation and indicates that there was a pulse or parity error in the receipt of the drive's state during the I/O operation.

See retry/recovery section for recovery details.

```
04068   CZUDC   HRD ERR  04068 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        UNKNOWN ERROR CODE DURING WRITE
        ERROR CODE RETURNED error_code
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

error_code: This is the error code returned to Test 4 by the UDA that Test 4 does not recognize.

The unknown error code occurs when the UDA returns an error code from an operation that Test 4 does not recognize. Possible UDA microcode change without Test 4 update.

See retry/recovery section for recovery details.

```
04069   CZUDC   HRD ERR  04069 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        UNKNOWN ERROR CODE DURING READ
        ERROR CODE RETURNED error_code
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

error_code: This is the error code returned to Test 4 by the UDA that Test 4 does not recognize.

The unknown error code occurs when the UDA returns an error code from an operation that Test 4 does not recognize. Possible UDA microcode change without Test 4 update.

See retry/recovery section for recovery details.

```
04070   CZUDC   SFT ERR  04070 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        TIMEOUT OF SEND
        command_type
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

command_type: See section following error 4078 for a description

If Test 4 tries to send a level 2 command to the drive, and receiver ready is deasserted, error 4070 occurs.

See retry/recovery section for recovery details.

```
04071   CZUDC   SFT ERR  04071 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        TIMEOUT OF RECEIVE
        command_type
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

command_type: See section following error 4078 for a description

This error is a failure of the drive to respond to an SDI level 2 command (see the SDI specification) before the drive-supplied command timeout expires.

See retry/recovery section for recovery details.

```
04072   CZUDC   SFT ERR  04072 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        FIRST WORD RECEIVED WAS NOT START FRAME
        command_type
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

command_type: See section following error 4078 for a description

The first word received by the UDA from the drive was not a valid message start frame.

See retry/recovery section for recovery details.

```
04073   CZUDC   SFT ERR  04073 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        FRAMING ERROR ON LEVEL 0 RECEIVE
        command_type
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                command_type:   See section following error 4078 for a description

                Error 4073 is caused by one or more of the following conditions:
                1) Illegal frame code -- the frame is not a message start, continue,
                or end frame.  2) Illegal sequence of frames -- such as a message
                start frame without ever receiving a message end frame.  This can be
                caused by the drive sending a response before the UDA asserts receiver
                ready, or a random hit on the SDI cable that garbles a frame or a bad
                drive transmitter or UDA receiver.

                See retry/recovery section for recovery details.


04074   CZUDC   SFT ERR  04074 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        CHECKSUM ERROR ON LEVEL 0 RECEIVE
        command_type
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                command_type:   See section following error 4078 for a description

                The checksum attached to a message end frame did not match the checksum
                computed over the level 2 command.  This could be caused by a bad drive
                transmitter, bad UDA receiver, incorrectly computed checksum by the
                drive (unlikely) or a random hit on the SDI cable.

                See retry/recovery section for recovery details.


04075   CZUDC   SFT ERR  04075 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
        DISK EXERCISER  DM PC:xxxx  UDA AT xxxxxx  DRIVE xxx  RUNTIME hh:mm:ss
        BUFFER SIZE SMALLER THAN RESPONSE
        command_type
        REAL TIME STATE 0003
        STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100

                command_type:   See section following error 4078 for a description

                The buffer size set aside for the response was not large enough for the
                response received.  This is caused by the drive sending a response that
                is incorrect for the request sent to the drive, or the drive sending some
                garbage with the response.

                See retry/recovery section for recovery details.
```

04076 CZUDC SFT ERR 04076 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED
command_type
EXPECTED RESPONSE expected_response
RESPONSE RECEIVED response_received
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

command_type: See section following error 4078 for a description

expected_response: This is the correct response (HEX) for the command.

response_received: This is the response received from the drive, (HEX) where a 7D is an unsuccessful response. Any other than a 7D for this value indicates a <<VERY>> sick drive.

This is caused by receiving an UNSUCCESSFUL response from the drive, or the drive sending some response other than the correct response for the request sent to the drive. See the contents of status for the unexpected response error (or reason).

See retry/recovery section for recovery details.

04077 CZUDC HRD ERR 04077 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
DRIVE NEVER DEASSERTED RECEIVER READY AFTER SEND
command_type
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

command_type: See section following error 4078 for a description

This is caused by the drive not seeing a command sent by the UDA. The drive must deassert receiver ready to acknowledge that it did see a command via the SDI. If the drive saw only part of the command, it would have marked the command as unsuccessful. But in this case, the drive did not see any of the command and is now waiting for a command to be sent from the UDA.

04078 CZUDC HRD ERR 04078 ON UNIT 00 TST 04 SUB 000 PC: xxxxxx
DISK EXERCISER DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hh:mm:ss
UNKNOWN ERROR CODE RETURNED FROM LEVEL 2 RECEIVE
command_type
ERROR CODE RETURNED error_code
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

command_type: See section following error 4078 for a description

error_code: This is the error code returned to Test 4 by the UDA that Test 4 does not recognize.

The unknown error code occurs when the UDA returns an error code from an operation that Test 4 does not recognize. Possible UDA microcode change without Test 4 update.

See retry/recovery section for recovery details.

NOTE: Errors 4070 - 4078 will become device fatals if attempted 3 times. If dropping of units are inhibited, error recovery is the same as if the error was a soft error.

command_type: in errors 4070-4078 command_type is one of the following level 2 commands:

ATTEMPTING TO BRING DRIVE ONLINE
ATTEMPTING TO ISSUE SEEK
ATTEMPTING TO GET STATUS
ATTEMPTING DRIVE CLEAR CMD
ATTEMPTING TO BRING DRIVE ONLINE
ATTEMPTING TO CHANGE MODE
ATTEMPTING ERROR RECOVERY CMD
ATTEMPTING TO ISSUE SEEK
ATTEMPTING TO RECALIBRATE

The following commands types occur only during initialization, and will cause a device fatal if they occur. Inhibiting the dropping of units has no effect on these errors.

ATTEMPTING TO SPIN UP DRIVE
ATTEMPTING TO GET COMMON CHAR
ATTEMPTING TO GET SUBUNIT CHAR

If <<ANY>> error occurs during initialization, <<NO>> testing is done on <<ANY>> drive attached to the UDA that the initialization erorr occured on. See error number 4016.

3.2.9 SPECIAL DEVICE FATAL (05000)

05000 CZUDC DVC FTL 05000 ON UNIT 00 TST 002 SUB 000 PC: xxxxxx
DISK zzzzzzzz DM PC:xxxx UDA AT xxxxxx DRIVE xxx RUNTIME hhh:mm:ss
UNABLE TO FIND REQUESTED DRIVE FOR TESTING
THE FOLLOWING IS VISIBLE ON THE PORTS
UDA PORT 0 -- description
UDA PORT 1 -- description
UDA PORT 2 -- description
UDA PORT 3 -- description

Where zzzzzzzz is either 'RESIDENT', 'FUNCION' or 'EXERCISER'. This message is presented when the specified drive was not found by test 2, test 3 or test 4 on any of the ports. A description of what was each port follows.

NO DRIVE ATTACHED
- There is nothing on the port. If there is suppose to be a drive on this port, make sure there is an odd number of cables between the UDA and the drive and make sure the cables are properly attached.

RCVR RDY NEVER ASSERTED
- The device on the port did not assert RCVR RDY while trying to get state.

TIMEOUT OF SEND
- Sending an SDI command timed out. RCVR RDY is not asserted.

TIMEOUT OF RECEIVE
- Receiving an SDI command timed out. The drive failed to respond to an SDI level 2 command before a timeout expired.

FIRST WORD RECEIVED WAS NOT START FRAME
- The first word received by the UDA from the drive was not a valid message start frame.

FRAMING ERROR ON LEVEL 0 RECEIVE
- The device and the UDA are out of sync or an illegal frame code (the frame is not a message start, continue, or end frame) or illegal sequence of frames. This can be caused by the drive sending a response before the UDA asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or UDA receiver.

CHECKSUM ERROR ON LEVEL 0 RECEIVE
- The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad UDA receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable.

RESPONSE LONGER THAN EXPECTED FOR CMD
- The buffer size set aside for the response was not large enough for the response received. This is caused by the drive sending a response that is incorrect for the request sent to the drive, or the drive sending some garbage with the response.

DRIVE n[, consecutive drive numbers if subunited drive] [further explanation]
- A drive was found at the end of the cable. It may be a subunited drive, so all the subunit numbers are printed. A further explanation may be presented. These further explanations are:

DRIVE NOT AVAILABLE TO THIS UDA
- The drive was found but is not available to this UDA. It may be dual ported and the drive is online to another controller.

UNSPINABLE DRIVE
- The drive is unspinable. The drive may be powered up but the RUN/STOP switch may be popped out.

3.3 TEST 4 RETRY/RECOVERY METHODS

ECC Error on Disk Read

ECC DETECTED ERROR, BUT CORRECTION FAILED
ECC CORRECTIONS EXCEED THRESHOLD
ECC DETECTED ERROR (If ECC correction disabled)

Retry/Recovery - The UDA or Test 4 will first re-read the sector with the erroneous ECC N times, then N times for each level of error recovery the drive supports. The value of N is an SDI drive characteristic. This retry mechanism will persist until either the recovery level reaches zero or the operation succeeds. It should be noted that the manual intervention questions can disable retries (in this case the recovery fails the first time) and disable error correction (i.e., no ECC correction will be performed). ECC correction and retries are <<ALWAYS>> enabled when the Test 4 is reading the RCT.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - Test 4 will issue a hard error for the sector. No soft errors will be counted.

Error Detecting Code (EDC) Error

EDC DETECTED ERROR BUT ECC DID NOT
ECC CORRECTION SUCCEEDED, BUT EDC DETECTS ERROR

This error is indicative of a UDA hardware error, either a SERDES failure or an undetected RAM failure, or a sector that was written with an incorrectly computed EDC.

Retry/Recovery - The UDA or Test 4 will re-read the sector with the erroneous EDC N times, then N times for each level of error recovery the drive supports. The value of N is an SDI drive characteristic. This retry mechanism will persist until either the recovery level reaches zero or the operation succeeds. It should be noted that the manual intervention questions can disable retries (in this case the recovery fails the first time). Retries are <<ALWAYS>> enabled when the Test 4 is reading the RCT.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - Test 4 will issue a hard error for the sector. No soft errors will be counted.

SDI Level 2 and Asynchronous Errors

The SDI level 2 errors are as follows:

- Packet acknowledge failure
- Level 2 command error response, "DE" bit set
- Level 2 command error response, "PE" or "RE" bit set
- Receipt of erroneous drive response
- Seek complete timeout
- Asynchronous drive errors

Level 2 errors are always retried, even if retries are disabled in the manual intervention questions.

In the following retry/recovery algorithms, the Test 4 'Generic error recovery' is the following steps:

1. Issue online command

2. Get status

   2a. If the port, run or spindle ready (PS, RU or SR) bit is deasserted, an Immediate device fatal error is reported and the unit and all its subunits are dropped from testing.

   2b. If the recalibrate requested (RR) bit is set, Test 4 will issue a RECALIBRATE, then SEEK <<AFTER>> generic error recovery is complete.

   2c. If the drive error (DE) bit is set, Test 4 will issue a SEEK <<AFTER>> generic error recovery is complete.

3. If no drive errors, go to 5

4. Send DRIVE CLEAR command

5. Change mode

NOTE: If the drive's timeout expires once, so the drive asserts attention just to get Test 4 to issue a level 2, Test 4 will go through the above error recovery. However, since the timeout expiring is not an error, no error message is issued.

Packet Acknowledge Failure

TIMEOUT OF SEND
TIMEOUT OF RECEIVE

The timeout of send occurs when the UDA attempts to send a level 2 command to the drive, but the drive's receiver ready is not asserted. Timeout of receive is a failure of the drive to respond to an SDI level 2 command (see the SDI specification) before the drive-supplied command timeout expires. These errors are grouped together because their recoveries are the same.

Retry/Recovery - UDA - The steps listed below are performed.

1. The drive is initialized.

2. An SDI GET STATUS command is issued.

3. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR command.

4. An SDI SEEK command is issued.

5. The command is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. The drive is initialized

2. Test 4 Generic error recovery is performed

3. An SDI SEEK command is issued.

4. The command is retried.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence will be repeated two times and, if the failure persists, the Test 4 will issue a device fatal error and the drive and all its subunits will be dropped. It should be noted that the retry strategy for SDI level 2 errors involves issuing additional level 2 commands. The retry count is the sum of all retries on all SDI level 2 commands, including those commands issued in recovery attempts.

Level 2 Command Error Response - "DE" Bit Set

RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED
SEEK RECEIVED UNSUCCESSFUL RESPONSE

An UNSUCCESSFUL response to a level 2 command, with the "DE" bit set in the status response, notifies the Test 4 that a drive error was detected (or occurred) in connection with the execution of the SDI command.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. The drive error is cleared by an SDI DRIVE CLEAR command and a SEEK command is issued for the cylinder where the drive was positioned when the error was reported.

3. The command is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed
Note that because the "DE" bit is set, Test 4 generic errcr recovery will issue a SEEK (see generic error recovery)

2. The command is retried

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if the failure persists, the Test 4 will issue a device fatal error and the drive and all its subunits will be dropped.
Note that the
retry strategy for SDI level 2 errors involves issuing additional level 2 commands. The retry count is the sum of all retries on all SDI level 2 commands, including those commands issued in recovery attempts.

Level 2 Command Error Response - "PE" or "RE" Bit Set

RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED
SEEK RECEIVED UNSUCCESSFUL RESPONSE

An UNSUCCESSFUL response to a level 2 command with the "PE" or "RE" bit set in the status response notifies the Test 4 that the command either was not appropriate for the state of the drive, or that the command contained invalid arguments.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued

2. The drive error is cleared by an SDI DRIVE CLEAR command.

3. The controller verifies the state of the drive and, if possible, retries the level 2 command. Otherwise, the UDA notifies the host and bypasses subsequent retries.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. The command is retried

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if the failure persists, the Test 4 will issue a device fatal error and the drive and all its subunits will be dropped. Note that the retry strategy for SDI level 2 errors involves issuing additional level 2 commands. The retry count is the sum of all retries on all SDI level 2 commands, including those commands issued in recovery attempts.

Receipt of an Erroneous Drive Response

FIRST WORD RECEIVED WAS NOT START FRAME
FRAMING ERROR ON LEVEL 0 RECEIVE
CHECKSUM ERROR ON LEVEL 0 RECEIVE
BUFFER SIZE SMALLER THAN RESPONSE
UNKNOWN ERROR CODE RETURNED FROM LEVEL 2 RECEIVE (hard error)

The first word not start frame error is caused when the UDA does not see a valid message start frame as the first frame received from the drive. The framing error is caused by the UDA receiving an illegal frame code -- the frame is not a message start, continue, or end frame or Illegal sequence of frames -- such as a message start frame without ever receiving a message end frame. The checksum error occurs when a message end frame checksum did not match the checksum computed over the level 2 command. The buffer size smaller than response error occurs when the buffer set aside for the response was not large enough for the response received. The unknown error code is returned when the UDA returns an error code that the Test 4 does not recognize. These errors are grouped together because their recoveries are the same.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR command.

3. The command is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. The command is retried

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if the failure persists, the Test 4 will issue a device fatal error and the drive and all its subunits will be dropped. Note that the retry strategy for SDI level 2 errors involves issuing additional level 2 commands. The retry count is the sum of all retries on all SDI level 2 commands, including those commands issued in recovery attempts.

Seek Complete Timeout

ATTN ASSERTED DURING SEEK -- ERROR OR LOGGABLE INFORMATION
SEEK DID NOT COMPLETE, NEITHER ATTN OR R/W RDY WAS ASSERTED

This error occurs when the drive fails to assert READ/WRITE READY, indicating the successful completion of a seek, or asserts the SDI ATTENTION signal without asserting the READ/WRITE READY signal, indicating the unsuccessful completion of a seek.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR command.

3. The SEEK is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. The SEEK is retried

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if the failure persists, the Test 4 will issue a device fatal error and the drive and all its subunits will be dropped. Note that the retry strategy for SDI level 2 errors involves issuing additional level 2 commands. The retry count is the sum of all retries on all SDI level 2 commands, including those commands issued in recovery attempts.

Asynchronous Drive Errors

ATTN ASSERTED UNEXPECTEDLY, ASYN DRIVE ERROR OR LOGGABLE INFORMATION -- THIS IS AN <<UNCOUNTED>> SOFT ERROR

Asynchronous drive errors are those errors reported by the drive which are not related to a level 2 or command. These errors are reported by the drive using the SDI ATTENTION signal. Examples are OFF CYLINDER and HDA OVERTEMPERATURE errors. Drive errors are reported to the controller by the "DE" or "WE" bit being set in the error byte in the status response.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. The drive error is cleared by an SDI DRIVE CLEAR command and, if the error is not "WE", a SEEK command is issued for the cylinder where the drive was last positioned.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. A SEEK is issued

NOTE: A "WE" is a write on a write protected drive; Test 4 detects this in a different manner, so "WE" will never be set.

Recovery Failure -

NOTE: There is a difference between the UDA in controller mode and the Test 4 for this type of error.

The UDA in controller mode will repeat the above sequence two times and, if the drive error persists, the drive would be marked as offlire.

Test 4 will <<NOT>> drop the drive after two retries. Instead, the drive will be dropped due to a side affect of such an error: A seek never completing, (causing a device fatal error) or Spindle ready dropping (causing a device fatal error).

Drive I/O Errors

The drive I/O errors occur either during the header compare process (i.e., before I/O actually begins) or during the I/O operation itself. They are as follows:

- Header not found
- Seek or head select error
- Data sync timeout
- Data or state clock timeout during operation (read/write)
- Receiver ready dropped during operation (read/write)
- Read/write ready dropped during operation (read/write)
- SERDES overrun error
- Drive failed to execute select track and (read/write)
- Real time state receive error

Header not found (header compare error)

HEADER NOT FOUND DURING (read/write)

This error occurs when the header compare routine fails to find the desired header (or a revectored version of the desired header) in two disk revolutions.

Retry/Recovery - UDA and Test 4 - Failure to find the desired header in two rotations of the disk will cause the Test 4 to search the Replacement and Caching Table (RCT) to check if the logical block number has been replaced. If a match is found, the Test 4 will perform the desired operation on the revectored block. Enabling/disabling retries has no affect on this operation.

Recovery success - No error is reported or counted.

Recovery Failure - A hard error (header not found) is reported.

Seek or head select error (Positioner Error)

SEEK OR HEAD SELECT ERROR DETECTED DURING (read/write)

This error occurs when the header comparison routine determines that the drive is positioned at the wrong cylinder and that the drive has not detected a seek error.

NOTE: The header comparison routine is active <<ONLY>> in the customer data area. This error will never be detected in the diagnostic area.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR command.

3. An SDI RECALIBRATE command is issued.

4. An SDI SEEK command is issued.

5. The I/O operation is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. An SDI RECALIBRATE command is issued.

3. An SDI SEEK command is issued.

4. If retries are disabled, Immediate recovery failure. Retries are <<ALWAYS>> enabled when the Test 4 is reading the RCT.

5. The I/O operation is retried.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if a drive I/O error persists, a hard error is reported for the sector. No soft errors are counted.

Data Sync Timeout Error

DATA SYNC TIMEOUT DURING READ

This error occurs on a read operation after the correct header has been found and the UDA times out waiting for the data sync word.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR COMMAND.

3. An SDI SEEK command is issued.

4. The read operation is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. An SDI SEEK command is issued.

3. If retries are disabled, Immediate recovery failure. Retries are <<ALWAYS>> enabled when the Test 4 is reading the RCT.

4. The read operation is retried.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if a drive I/O error persists, a hard error is reported for the sector. No soft errors are counted.

Data or state clock timeout (Loss of Drive Clock)
Receiver ready failure (Loss of Drive Receiver Ready)

DATA OR STATE CLOCK TIMEOUT DURING (read/write)
RCVR RDY DROPPED DURING (read/write)
COULD NOT SEND SELECT TRACK AND (read/write) CMD OR
HEADER SYNC TIMEOUT WITH INVALID STATE

The loss of drive clock occurs when the UDA is clocking data to or from the drive through the SERDES. Failure of a word to be clocked in during a 125 millisecond time period triggers a loss of drive clock error. The loss of drive receiver ready is detected when the UDA is trying to send out a real-time read or write command. Unable to select track and read or write occurs when the UDA attempts to send the select track and read/write level 1 cmd, but receiver ready is deasserted or the state is invalid so it cannot send the command (the SERDES could also be broken so it's unable to send the command). The same error is generated if the UDA gets a header sync timeout, and when it looks at the drive's state, it is either invalid or reciever ready is deasserted (header sync timeout is <<NOT>> a error -- it's quite normal on a high-density disk). These errors are grouped together because their recoveries are the same.

Retry/Recovery - UDA - The steps listed below are performed.

1. The drive is initialized.

2. An SDI GET STATUS command is issued.

3. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR command.

4. An SDI SEEK command is issued.

5. The I/O operation is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. The drive is initialized

2. Test 4 Generic error recovery is performed

3. An SDI SEEK command is issued.

4. If retries are disabled, Immediate recovery failure. Retries are <<ALWAYS>> enabled when the Test 4 is reading the RCT.

5. The I/O operation is retried.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if a drive I/O error persists, a hard error is reported for the sector. No soft errors are counted.

Read/Write ready dropped (Loss of Drive Read/Write Ready)
SERDES Overrun Error
Real Time State Receive Error (Real Time Drive State Receive Error)

R/W RDY DROPPED DURING (read/write)
SERDES OVERRUN ERROR DURING READ
REAL TIME STATE RECEIVE ERROR DURING (read/write)
UNKNOWN ERROR CODE DURING (read/write)

The loss of read/write ready error is detected either before an I/O has begun when trying to send out the real time command or at the end of an I/O operation when checking for errors. The SERDES overrun error is detected on a read operation and is indicative of a drive whose transfer rate is greater than 23 MHZ or a broken SERDES. The real time drive state receive error is detected at the end of an I/O operation and indicates that there was a pulse or parity error in the receipt of the drive's state during the I/O operation. The unknown error code is returned when the UDA returns an error code that the Test 4 does not recognize. They are grouped together because their recoveries are the same.

Retry/Recovery - UDA - The steps listed below are performed.

1. An SDI GET STATUS command is issued.

2. If the status obtained in the previous step indicated error conditions, these error conditions are resolved and then cleared by an SDI DRIVE CLEAR command.

3. An SDI SEEK command is issued.

4. The I/O operation is retried.

Retry/Recovery - Test 4 - The steps listed below are performed.

1. Test 4 Generic error recovery is performed

2. An SDI SEEK command is issued.

3. If retries are disabled, Immediate recovery failure. Retries are <<ALWAYS>> enabled when the test 4 is reading the RCT.

4. The read operation is retried.

Recovery success - One soft error is counted for the entire operation including retries.

Recovery Failure - The above sequence is repeated two times and, if a drive I/O error persists, a hard error is reported for the sector. No soft errors are counted.

3.4 DEC STANDARD 166 EXCERPTS

3.4.1 THE REPLACEMENT AND CACHING TABLES

The Replacement and Caching Tables record the locations of all revectored LBN sectors and the status of each RBN on the unit. Each copy of the table is organized in ascending RBN order, with an entry for each RBN sector on the unit. There are "n" copies of the table on the unit, where "n" is a device characteristic. The tables are stored at the high address end of the LBN area of the unit. Table entries (and RBNs) are allocated via a hash algorithm described later.

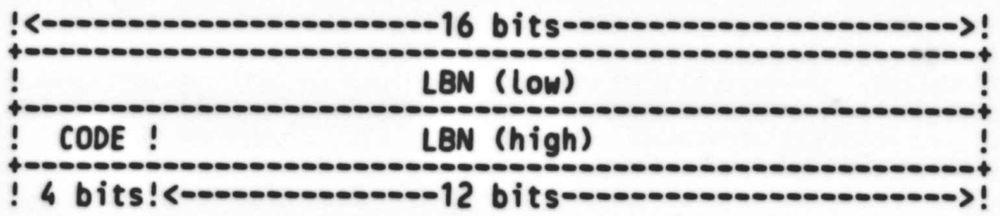
Replacement And Caching Table Format -

Each entry in the Replacement and Caching Table represents an RBN on the unit. The table is ordered in ascending RBN order. Thus the first entry corresponds to the first RBN on the unit, etc. The size of each copy of the table may exceed that required to contain an entry for each RBN on the unit since additional entries may be required to align the table so that adjacent copies can begin on a track boundary. Entries that do not correspond to RBNs on the unit are called "null entries"; there is always at least one null entry at the end of the RCT. All other entries past this last null entry are undefined.

NOTE

The RCT pad area is controller specific and should never be accessed by the host.

The format of a replacement block descriptor in the Replacement and Caching Tables is:



Where:

LBN is the Logical Block Number of a revectored LBN sector.

CODE is one of the following octal values:

00 - Unallocated (empty) replacement block.

02 - Allocated replacement block - primary RBN.

03 - Allocated replacement block - non-primary RBN.

04 - Unusable replacement block.

* 05 - Alternate unusable replacement block

10 - Null entry - no corresponding RBN sector.

For codes 00, 04, and 10 the LBN field is always zero.

NOTE

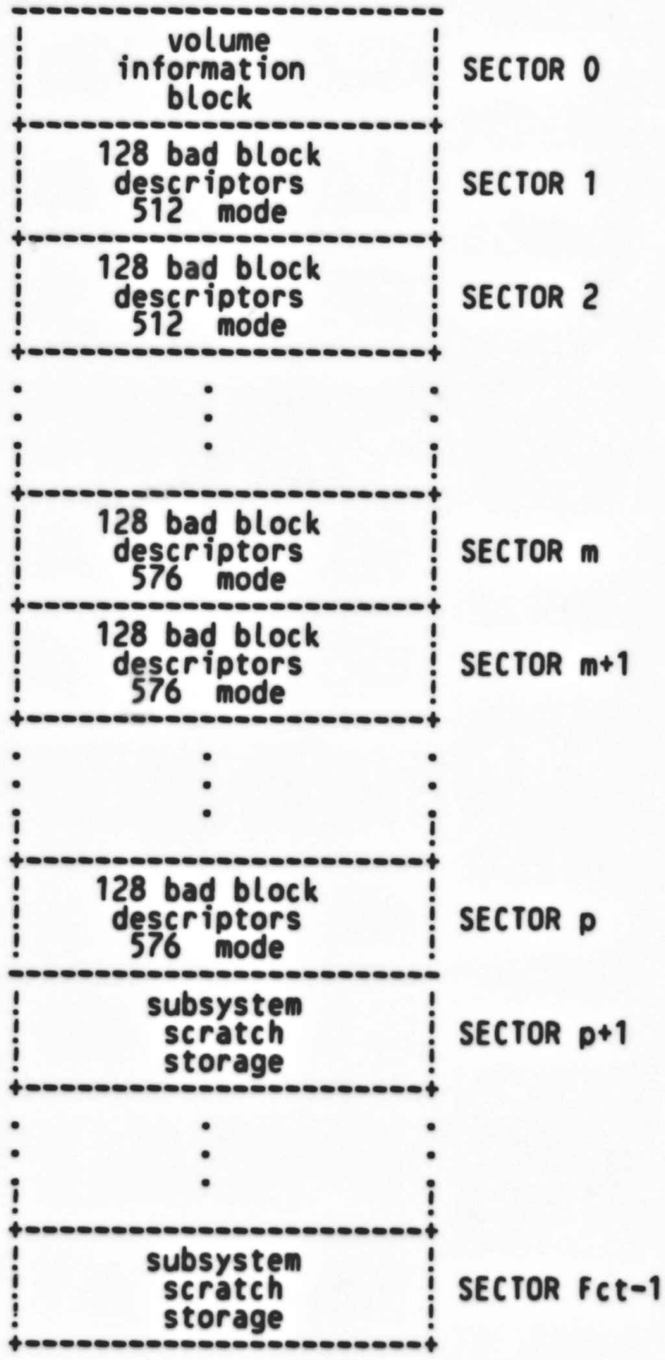
* This code is reserved. Programs should treat this code as if it were code 04.

Embedded-controllers with no distinction between primary and secondary RBN's must use:

1. Code 02 if the replacement block can be retrieved with little degradation of performance for all blocks.
2. Code 03 if accessing the replacement block has a large impact on performance for all blocks.

3.4.2 FCT Structure

Each copy of the FCT is composed of one volume information block, one 512 byte format table, one 576 byte format table, and one subsystem temporary storage area (distributed amongst the alignment pads). An FCT copy has the following format:



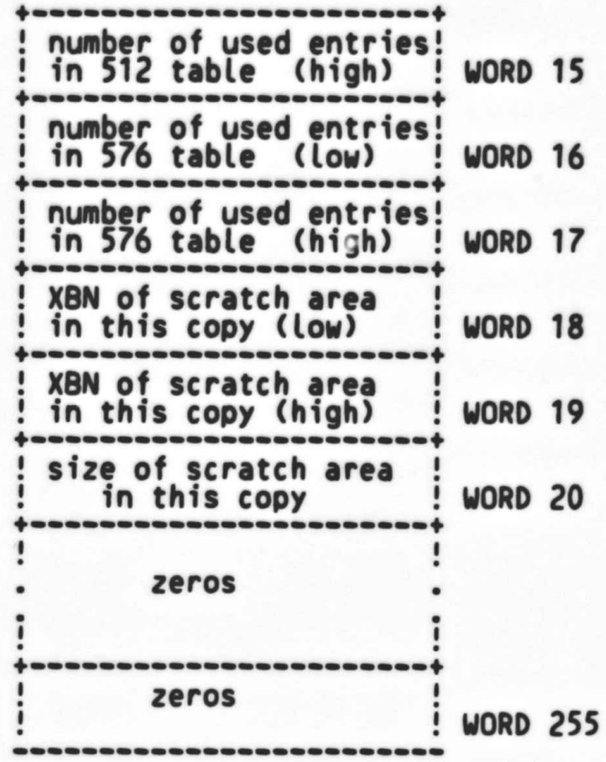
The XBN area itself is always formatted to contain 512 byte sectors. The calculations for m and p are:

m := ((((Lc*g*t*r)+1)/2)+127)/128

p := 2*m

Sector 0 contains various volume identification information. The format is:

media mode	WORD 0
formatting instance number	WORD 1
volume serial number least significant word	WORD 2
volume serial number	WORD 3
volume serial number	WORD 4
volume serial number most significant word	WORD 5
date that volume was first formatted (low)	WORD 6
date that volume was first formatted	WORD 7
date that volume was first formatted	WORD 8
date that volume was first formatted (high)	WORD 9
date of most recent volume formatting (low)	WORD 10
date of most recent volume formatting	WORD 11
date of most recent volume formatting	WORD 12
date of most recent volum formatting (high)	WORD 13
number of used entries in 512 table (low)	WORD 14



Where:

WORD 0: "Media Mode" - is "126736" for a 512 byte format and "074161" for a 576 byte format. During formatting the media mode word is set to zero.

4.0 PERFORMANCE AND PROGRESS REPORTS

At the end of each pass, the pass count is given along with the total number of errors reported since the diagnostic was started. The "EOP" switch can be used to control how often the end of pass message is printed. Section 2.2 describes switches.

A statistical report will automatically be printed periodically (approximately every fifteen minutes) and at the end of test #4. It can be suppressed by setting the Inhibit Statistical Report flag (e.g. START/FLAGS:ISR). This is the same report that can be printed on demand with the PRINT command.

During tests 1, 2, and 3, the report will look like the following example:

```
TEST 1 IN PROGRESS    RUN TIME 2:24:10
```

During test #4, the report will contain statistics on each drive for the current pass of the test; for example:

```
TEST 4 IN PROGRESS    RUN TIME 2:24:10

UNIT DRIVE     SERIAL-NUMBER SEEKS MBYTES  MBYTES  HARD   SOFT     ECC
                             X1000  READ  WRITTEN ERRORS ERRORS
   0     0            1002     12     36       22      0      0       1
   1     4      7342102112     14     42       29      0      2       0
```

Explanation of each column:

UNIT	The unit number (number of HW P-table).
DRIVE	The drive number (the number which appears on the "unit plug" on the front of the disk drive).
SERIAL-NUMBER	The decimal serial number of the disk drive.
SEEKS X1000	The decimal number of seeks performed by this drive during this pass of test 4. Multiply value by 1000.
MBYTES READ	The number of mega-bytes (million bytes) read by this drive during this pass of test 4. It is this value that is used to optionally drop a drive by the READ TRANSFER LIMIT software question.
MBYTES WRITTEN	The number of mega-bytes written by this drive during this pass of test 4.
HARD ERRORS	The number of hard error reports printed for this drive during this pass of test 4. It is this value that is used to optionally drop a drive by the ERROR LIMIT software question.

SOFT ERRORS	The number of soft errors reported for the drive during this pass of test 4. A soft error is any error condition that resulted in a retry operation that eventually succeeded in recovering from the error condition. One soft error is counted even though several retry attempts may be made and does not correspond to the number of soft error reports printed. To see the soft error reports, you must change the default answer to the SUPPRESS PRINTING SOFT ERRORS software question.
ECC	The number of times data read from the drive was modified using the error correction code (ECC) and resulted in a matching error detection code (EDC).

5.0 TEST SUMMARIES

The UDA Host Resident Diagnostic consists of one PDP-11 diagnostic supervisor program that runs in the PDP-11 processor and four programs that run in the UDA's buffer memory through an interpreter called the "diagnostic machine" which resides in the UDA. The PDP-11 program mainly is responsible for downline loading the "diagnostic machine" programs into the UDA and starting their execution. The "diagnostic machine" program controls the testing from that point by requesting the PDP-11 processor to supply information, print error messages and update statistics. The "diagnostic machine" program informs the PDP-11 processor when a test is complete.

Eight "diagnostic machine" programs are in the ZUDDC0.PAK data file which is read from the XXDP+ system device by the PDP-11 program. Two copies of each of the four "diagnostic machine" programs are included. One copy is automatically selected for UDA-50s with the M7161 and M7162 modules, while the other copy is selected for UDA-50s with the M7485 and M7486 modules. The data file comes with listings of each program.

5.1 TEST # 1 - UNIBUS ADDRESSING TEST

The purpose of test #1 is to complete the testing of the Unibus interface in the UDA. The UDA resident diagnostic is not able to completely test the Unibus interface because communication with the PDP-11 processor is necessary. Specifically, this test will:

1. Check that every address line on the Unibus can be driven to both one and zero states.
2. Check that the UDA can interrupt the PDP-11 processor at the proper priority level and vector.
3. Exercise the Unibus interface by transferring blocks of data to and from Unibus memory.

This test assumes that the following are being tested by the UDA Resident Diagnostic:

1. All data bits can be written and read correctly.
2. NPR cycles can be executed correctly.

Test 1 is divided into six subtests. One at a time, each UDA selected for testing will run each subtest.

Subtest 1 makes sure that the UDAIP and UDASA registers are existant and runs the first part of the UDA's resident diagnostics.

Subtest 2 initializies the UDA into diagnostic loop mode.
In this mode any value written into the UDASA is echoed in the UDASA.

In subtest 3, the UDA is initialized with interrupts enabled. The vector address and priority level will be determined solely from the answers to the hardware questions. If the hardware vectors to the wrong address, it is impossible to determine the result. A descriptive error message of the problem will not occur (the program or processor may hang or an unrelated message may occur). Therefore, the message 'TESTING INTERRUPT ABILITY OF UDA AT ADR xxxxxx VEC xxx...'' isprinted just before the UDA is requested to cause an interrupt and the word ''COMPLETED'' is printed (on the same line) when the interrupt test is completed. If the word ''COMPLETED'' does not follow the first message, it should be apparent that the interrupt caused the diagnostic or processor to go astray. The priority level of the interrupt request is also verified.

Subtest 4 and 5 initializes the UDA using different sizes of the host communications area. The different sizes of the host communications area are supplied to allow the UDA Resident Diagnostic to do the most Unibus address testing possible. Interrupts are disabled. Any UDA Resident Diagnostic errors will be reported. Subtest 4 initializes the UDA with the smallest ring buffer size possible. Subtest 5 initializes the UDA with a large ring buffer area.

Subtest 6 downline loads a ''diagnostic machine'' program into the UDA. The ''diagnostic machine'' program is downline loaded from the memory space included in the host communications area when the UDA was first initialized. The UDA Resident Diagnostic has already verified that it can access these memory addresses, so the downline load command should perform properly. The ''diagnostic machine'' program is then started.

The ''diagnostic machine'' program asks the PDP-11 program to fill free memory (that memory available to the PDP-11 program that is not being used by the program or the Runtime Services) with an addressing pattern and report the location and size of the free memory. Every location of free memory is read and the data checked. Then, one by one, each address line is tested as follows:

1. Determine a test address by taking the first address of free memory and complimenting the address bit to be tested.
2. Read from the test address.
3. If a non-existant memory error occurs, the test is complete.
4. Write all ones to the first address of free memory then read from the test address. If data read is not all ones, then test is complete.
5. Write zeros to the first address of free memory then read from the test address. If data read is not zeros, then test is complete.
6. Report Unibus addressing error.

When all address bits have been tested, then block transfers to and from memory are tested with different data patterns. This data is transferred at the rate disk data is transferred to and from memory during normal UDA operation.

The next UDA selected for testing is then be tested in the same manner. When all UDAs have been tested, test #1 ends.

5.2 TEST # 2 - DISK RESIDENT DIAGNOSTIC TEST

The purpose of test #2 is to execute the diagnostics that run in each disk drive. These diagnostic programs may be resident in the disk drive or require downline loading from the ZUDDC0.PAK data file. (There currently are no disk drives that require downline loading and no such files exist in the ZUDDC0.PAK file. This program is designed such that they can be easily added in a future release.) This UDA diagnostic program only knows the procedure to execute the disk resident diagnostics and how to determine whether a test passed or failed.

One at a time, each UDA selected for testing is initialized and a "diagnostic machine" program downline loaded. The "diagnostic machine" program asks what drives are to be tested, then issues several commands to the disk drive and check for the correct response from the drive. This should serve as a good indicator that the UDA and disk drive can communicate.

A DIAGNOSE command is then issued to the drive to request the drive run all of its diagnostics. If the disk drive requests a downline load of a drive diagnostic, the diagnostic program is read from the XXDP+ load device, downline loaded into the disk drive and started. There is no limit to the number of downline loads that can be requested by a drive.

If the "Manual Intervention Mode" software question was answered "N" (default) testing proceeds to the next drive. When all drives on the UDA have been tested, the next UDA selected for testing is tested in the same manner. When all UDA's have been tested, test #2 ends.

If the "Manual Intervention Mode" software question was answered "Y", an interactive mode is entered to allow the operator to perform diagnostic activities on the disk drive as desired. The Service Manual for the disk drive must be used to determine what diagnostic capabilities are available.

```
First, a brief description of available commands is printed as
follows:

TEST #2 MANUAL INTERVENTION ON UNIT xx UDA AT xxxxxx DRIVE xxx
TO WRITE AND READ MEMORY:
  W DATA REGION OFFSET
  R REGION OFFSET
TO RUN A DIAGNOSTIC:
  D REGION
TO EXIT QUESTIONING:
  E
DATA, REGION AND OFFSET ARE HEX VALUES.
 ?

Commands may be typed after the question mark prompt. Each command
is processed as entered and results displayed immediately. The exit
command will allow the diagnostic to proceed.

Read and write commands remember the region and offset values.
Successive read and successive write commands automatically increment
to the next offset if the region and offset values are not typed. If a
region is typed but not an offset, offset zero is used.

Examples:

1.      W FF FFFC 4
2.      W 02
3.      R FFFC 4
          FFFC 0004/ FF
4.      R
          FFFC 0005/ 02
5.      W 21 FFFC
6.      R
          FFFC 0000/ 21

Command 1 writes one byte (FF) into region FFFC, offset 4. Command 2
writes one byte (02) into the next byte - region FFFC, offset 0005.
Commands 3 and 4 read the bytes back.  Command 5 writes one byte (21)
into the first byte of region FFFC.  Command 6 reads back that byte.

The diagnose command remembers the region from previous diagnose
commands only, because the region containing the diagnostic is
generally not the same region used to write parameters or read
results. If the diagnostic returns any data, the data is printed
immediately.
```

5.3 TEST # 3 - DISK FUNCTION TEST

The purpose of test #3 is to functionally test the disk drive. On a drive that is well diagnosed by its disk resident diagnostics (executed by test #2) these functional tests will have little value. On a drive that has no or minimal resident diagnostics, these functional tests will have more value.

Test #3 starts by initializing each UDA selected for testing and then downline loading a "diagnostic machine" program into each UDA. Once all UDAs have been started, the PDP-11 program responds to requests from all UDAs. When all the UDAs have indicated the end of testing, test #3 ends.

The "diagnostic machine" program performs the following functions on each drive:

1. Issue a DRIVE CLEAR command.
2. Issue RECALIBRATE command.
3. Issue a CHANGE MODE command to enable diagnostic cylinder access, set the drive to 512 byte sector size, and write protect.
4. Issue INITIATE SEEK command to last diagnostic cylinder.
5. Read all factory formatted sector headers. If no headers on a track can be read, report the error, otherwise continue.
6. Starting with cylinder 0, group 0 and incrementing through every cylinder on the disk, seek to a group, read a header on track 0 and then seek to the factory formatted diagnostic cylinder. Read from the diagnostic cylinder to verify disk positioned correctly.
7. Attempt to write on the first diagnostic cylinder while write protected.
8. Issue a CHANGE MODE command to enable formatting operations and disable write protect.
9. Format all writable DBNs in 512 byte format.
10. Write and read several data patterns to each writable DBN. Report an error if all DBNs on one track have an error.
11. Send invalid SDI level 2 and level 1 commands and check the results.
12. Go to the XBN area and read a copy of the FCT. Check to see if the drive has been properly formatted in 512 byte mode.
13. Issue a DISCONNECT command.

5.4 TEST # 4 - DISK EXERCISER

The purpose of test #4 is to exercise the disk drives in a manner similar to normal usage under standard operating systems. Execution of this test should give an indication of the performance of the disk drive. This test may be run for long or short periods of time, depending on how the software questions are answered.

These are two modes of operation for test #4:

1. Default operation on the entire area selected (customer or diagnostic) with all parameters selected for random operation as shown by default answers below.

2. Manual intervention mode where a number of questions are asked and operation is controlled by their answers.

Which mode is entirely determined by the answer to the first software question asking, "Enter manual intervention mode for special diagnosis?" This question would normally have been answered "N" (default) and testing will begin immediately. If answered "Y", the following series of questions will be asked for each unit selected for testing:

THE FOLLOWING QUESTIONS REFER TO UNIT xx UDA AT xxxxxx DRIVE xxx

This message will identify to which drive the questions are being asked. The entire series of questions will be asked for each drive, there is no short way to answer like in the hardware questions.

NUMBER OF BAD BLOCKS (D) 0 ?

An answer in the range of 1 to 16 will allow that many bad block numbers to be entered. The program will allow writes and reads to these blocks but no error messages will be printed for these blocks. Errors encountered on these blocks will not appear in the statistics. Answer zero to bypass entering bad blocks.

BAD BLOCK (A) ?

This question will be asked the number of times requested by the previous answer. Any decimal number that can be converted into a 28-bit binary value will be accepted. No other error checking will be made at this time to determine if the block number actually exists on the disk.

DO YOU WANT TO CHANGE TESTING PARAMETERS FOR THIS DRIVE (L) N ?

Answer "N" to bypass all further questioning on this drive.
Answer "Y" to be asked the following questions.

READ ONLY (L) N ?

Answer "Y" to dictate read only and prevent test #4 from performing any writes to the disk.

WRITE ONLY (L) N ?

This question will only be asked if the previous question was answered "N". Answer "Y" to dictate write only.

CHECK ALL WRITES BY READING (L) N ?

Answer "Y" to cause all writes to be checked by reading the data immediately after the write operation.

RANDOMLY CHECK WRITES BY READING (L) Y ?

This question will only be asked if the previous question was answered "N". Answer "Y" for the write check to be performed randomly. Answer "N" if write checks are not desired. This question is asked no matter how previous questions were asked.

DATA PATTERN - 0 FOR RANDOM SELECTION (D) 0 ?

There are 16 data patterns available, selected as 1 to 16. Pattern number 0 will cause patterns 1 to 15 to be randomly selected for each write. If pattern number 16 is selected, the following set of questions will be asked for a pattern to be input.

ENABLE ECC DATA CORRECTION (L) Y ?

A "Y" answer will enable the use of ECC to correct data errors. If the number of corrections is within the drive's threshold, an informational message will be printed identifying the block number. These ECC corrections will also appear in the statistical report for the drive.

An "N" answer will prevent the use of ECC. All ECC errors will cause an error message to be printed and retries to be attempted.

COMPARE ALL DATA READ (L) N ?

Answer "Y" to cause a data compare after every read.

RANDOMLY COMPARE DATA READ (L) Y ?

This question will only be asked if the previous question was answered "N". Answer "Y" for the data compare to be performed on random records. Answer "N" if data compares are not desired.

```
ENABLE RETRIES (L) Y

        A "Y" answer will enable retries to be performed on disk
        errors.

RANDOM ACCESS MODE (L) Y ?

        Answer "Y" to cause block numbers to be chosen randomly.
        Answer "N" to cause block numbers to be selected sequentially
        up and down the disk surface.

DO YOU WISH TO:
   0 - TEST ENTIRE AREA SELECTED
   1 - SPECIFY BEGIN/END SETS TO TEST
   2 - SPECIFY TRACKS AND CYLINDERS TO TEST
   3 - SPECIFY GROUPS AND CYLINDERS TO TEST
   4 - SPECIFY CYLINDERS TO TEST
(D) 0 ?

        This question specifies the options available to limit testing
        to a portion of the selected area (customer or diagnostic) of
        the disk.  A zero answer is the default which specifies to use
        the entire area for the test.  Other answers will cause
        additional questions to be asked.

NUMBER OF BEGIN/END SETS (D) 1 ?
BEGIN BLOCK (A) 0 ?
END BLOCK (A) 0 ?

        These questions are asked if begin/end sets were selected to
        limit the testing area (Answer 1).  One to four sets may be
        specified.  The BEGIN BLOCK and END BLOCK questions are asked
        as many times as needed.

NUMBER OF TRACKS TO TEST (D) 1 ?
TRACK (D) 0 ?

NUMBER OF GROUPS TO TEST (D) 1 ?
GROUP (D) 0 ?

        One of these sets of questions is asked if either tracks and
        cylinders or groups and cylinders was specified to limit the
        testing area (Answers 2 or 3).  Up to seven tracks or groups
        may be specified on which testing will be limited.

DO YOU WISH TO LIMIT THE CYLINDERS TESTED (L) N ?

        This question is asked only after the tracks or groups have
        been specified above.  If testing is to be further limited to
        a set of cylinders, answer "Y" and the following two ques-
        tions will be asked:
```

```
STARTING CYLINDER (A) 0 ?
ENDING CYLINDER (A) 0 ?
```

These questions are asked if the question immediately above was answered "Y" or if cylinders were selected to limit the testing area (Answer 4). One set of cylinder numbers may be specified to limit the testing area.

After the above questions have been asked for all drives selected for testing, the following questions will be asked if data pattern 16 was selected for any drive:

```
NUMBER OF WORDS IN DATA PATTERN 16 (D) 1 ?
DATA WORD (O) 0 ?
```

Data pattern 16 can be input by these questions. A data pattern consists of a buffer of one to 16 words which is repeated throughout the data portion of the disk block. Enter the contents of the data pattern buffer. The DATA WORD question will be repeated as needed.

Test #4 will then initialize each UDA selected for testing and downline load a "diagnostic machine" program into each UDA.
Becaus the "diagnostic machine" programs are too large to fit both copies in memory at the same time (as done in Tests 1 through 3), the program checks which type of UDA-50s are being tested. If all are of the same type, that program is read. If both types are selected for testing, the program for the UDA-50 with the M7485 and M7486 boards is read first, then the program for the UDA-50 with the M7161 and M7162 boards. Error 00009 is printed declaring an illegal configuration if the XXDP+ load device is connected to one of the first type of UDA-50s being tested (M7485-6 boards).
The "diagnostic machine" program asks what drives are to be tested and then for the parameters for each drive (the answers to the manual intervention questions or their defaults). Once all UDAs have been started, the PDP-11 program responds to requests from all UDAs.

The disks are then be exercised according to the parameters. The exercise consists of selecting a disk sector, seeking to the proper cylinder, then reading or writing the sector. The parameters control how the disk sector is selected, whether the sector is written or read and whether a write is followed by a read (write check).

The "diagnostic machine" program periodically sends statistics to the PDP-11 program. These statistics include counts of reads, writes, seeks and errors on a per drive basis. The PDP-11 program accumulates the statistics from all the UDAs and watches for the transfer limit to be exceeded. As long as the error log is not enabled, the exceeding of the transfer limit will cause the end of test #4.

Each time an error occurs, the "diagnostic machine" tells the PDP-11 program. A message is printed (or stored in the log buffer) and then the error limit for the drive is checked. If the error limit has been reached, the drive is dropped from testing. If no more drives remain to be tested, test #4 will end (unless the error log is enabled).

When the end of test #4 occurs, the accumulated statistics for each drive is printed. This statistical report can be printed at any time during test #4 by typing control-C then the PRINT command.

The data patterns used by test #4 are indicated below. Each pattern is generated by writing the pattern number in each 4-bit nibble of the first word, then repeating the data pattern (sequence of one to 16 words) throughout the rest of the data buffer. Pattern number 16 writes nibbles of zeros. When pattern number zero is used, the actual pattern number written (1 to 15) is placed in the nibbles.

PATTERN 0 This pattern number is used to indicate any pattern number 1 to 15 chosen at random.

PATTERN 1 Words in pattern sequence - 1

Sequence (Octal) 105613
Sequence (Hex) 8B8B

PATTERN 2 Words in pattern sequence - 1

Sequence (Octal) 031463
Sequence (Hex) 3333

PATTERN 3 Words in pattern sequence - 1

Sequence (Octal) 030221
Sequence (Hex) 3091

PATTERN 4 Words in pattern sequence - 16 (Shifting ones)

Sequence (Octal) 000001, 000003, 000007, 000017, 000037, 000077, 000177, 000377, 000777, 001777, 003777, 007777, 017777, 037777, 077777, 177777

Sequence (Hex) 0001, 0003, 0007, 000F, 001F, 003F, 007F, 00FF, 01FF, 03FF, 07FF, 0FFF, 1FFF, 3FFF, 7FFF, FFFF

PATTERN 5 Words in pattern sequence - 16 (Shifting zeros)

Sequence (Octal) 177776, 177774, 177770, 177760, 177740,
177700, 177600, 177400, 177000, 176000,
174000, 170000, 160000, 140000, 100000,
000000

Sequence (Hex) FFFE, FFFC, FFF8, FFF0, FFE0, FFC0,
FF80, FF00, FE00, FC00, F800, F000,
E000, C000, 8000, 0000

PATTERN 6 Words in pattern sequence - 16

Sequence (Octal) 000000, 000000, 000000, 177777, 177777,
177777, 000000, 000000, 177777, 177777,
000000, 177777, 000000, 177777, 000000,
177777

Sequence (Hex) 0000, 0000, 0000, FFFF, FFFF, FFFF,
0000, 0000, FFFF, FFFF, 0000, FFFF,
0000, FFFF, 0000, FFFF

PATTERN 7 Words in pattern sequence - (BINARY 1011011011011001)

Sequence (Octal) 133331
Sequence (Hex) B6D9

PATTERN 8 Words in pattern sequence - 16

Sequence (Octal) 052525, 052525, 052525, 125252, 125252,
125252, 052525, 052525, 125252, 125252,
052525, 125252, 052525, 125252, 052525,
125252

Sequence (Hex) 5555, 5555, 5555, AAAA, AAAA, AAAA,
5555, 5555, AAAA, AAAA, 5555, AAAA,
5555, AAAA, 5555, AAAA

PATTERN 9 Words in pattern sequence - 1 (BINARY 1101101101101100)

Sequence (Octal) 155554
Sequence (Hex) DB6C

PATTERN 10 Words in pattern sequence - 16

Sequence (Octal) 026455, 026455, 026455, 151322, 151322,
151322, 026455, 026455, 151322, 151322,
026455, 151322, 026455, 151322, 026455,
151322

Sequence (Hex) 2D2D, 2D2D, 2D2D, D2D2, D2D2, D2D2,
2D2D, 2D2D, D2D2, D2D2, 2D2D, D2D2,
2D2D, D2D2, 2D2D, D2D2

PATTERN 11 Words in pattern sequence - 1 (BINARY 0110110110110110)

Sequence (Octal) 066666
Sequence (Hex) 6DD6

PATTERN 12 Words in pattern sequence - 16 (Ripple one)

Sequence (Octal) 000001, 000002, 000004, 000010, 000020,
000040, 000100, 000200, 000400, 001000,
002000, 004000, 010000, 020000, 040000,
100000

Sequence (Hex) 0001, 0002, 0004, 0008, 0010, 0020,
0040, 0080, 0100, 0200, 0400, 0800,
1000, 2000, 4000, 8000

PATTERN 13 Words in pattern sequence - 16 (Ripple zero)

Sequence (Octal) 177776, 177775, 177773, 177767, 177757,
177737, 177677, 177577, 177377, 176777,
175777, 173777, 167777, 157777, 137777,
077777

Sequence (Hex) FFFE, FFFD, FFFB, FFF7, FFEF, FFDF,
FFBF, FF7F, FEFF, FDFF, FBFF, F7FF,
EFFF, DFFF, BFFF, 7FFF

PATTERN 14 Words in pattern sequence - 3

Sequence (Octal) 155555, 133333, 155555
Sequence (Hex) DB6D, B6DB, DB6D

PATTERN 15 Words in pattern sequence - 16

Sequence (Octal) 133331, 133331, 133331, 155554, 155554,
155554, 133331, 133331, 155554, 155554,
133331, 155554, 133331, 155554, 133331,
155554

Sequence (Hex) B6D9, B6D9, B6D9, DB6C, DB6C, DB6C,
B6D9, B6D9, DB6C, DB6C, B6D9, DB6C,
B6D9, DB6C, B6D9, DB6C

PATTERN 16 This is the operator selectable pattern in manual intervention mode. Questions are asked when test #4 is started for the operator to input the number of words in the sequence and the contents of the words.

Sample of terminal dialogue going through manual intervention questions:

```
DR>STA/TEST:4

CHANGE HW (L)  ? N

CHANGE SW (L)  ? Y

ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS (L) N ? Y

REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY

ERROR LIMIT (D)  32 ?
READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT (D)  0 ?
SUPPRESS PRINTING SOFT ERRORS (L) Y ? N
DO INITIAL WRITE ON START (L) Y ?
ENABLE ERROR LOG (L) N ?

THE FOLLOWING QUESTIONS REFER TO UNIT 0 UDA AT 172150 DRIVE 0

NUMBER OF BAD BLOCKS (D) 0 ? 2

BAD BLOCK (A) ? 234

BAD BLOCK (A) ? 8900

DO YOU WANT TO CHANGE TESTING PARAMETERS FOR THIS DRIVE (L) N ? Y

READ ONLY (L) N ?

WRITE ONLY (L) N ?

CHECK ALL WRITES BY READING (L) N ? Y

DATA PATTERN - 0 FOR RANDOM SELECTION (D) 0 ? 1

ENABLE ECC DATA CORRECTION (L) Y ?

COMPARE ALL DATA READ (L) N ? Y

ENABLE RETRIES (L) Y ?

RANDOM ACCESS MODE (L) Y ? N

DO YOU WISH TO:
   0 - TEST ENTIRE AREA SELECTED
   1 - SPECIFY BEGIN/END SETS TO TEST
   2 - SPECIFY TRACKS AND CYLINDERS TO TEST
   3 - SPECIFY GROUPS AND CYLINDERS TO TEST
   4 - SPECIFY CYLINDERS TO TEST
 (D) 0 ? 1

NUMBER OF BEGIN/END SETS (D) 1 ?

BEGIN BLOCK (A) 0 ?
```

```
END BLOCK (A) 0 ? 200
NUMBER OF WORDS IN DATA PATTERN 16 (D) 1 ?
DATA WORD (O) 0 ?



&
```

```
     1                          .SBTTL  PROGRAM HEADER
    32
    33 002000                           BGNMOD
    34                          ;++
    35                          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
    36                          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
    37                          ;--
    38
    39 002000                           POINTER BGNRPT, BGNSW, BGNSFT, ERRTBL, BGNSETUP
    40
    42 002000                   HEADER  CZUDC,C,0,0,1,PRIO7     ;FIELD SERVICE
       002000                                                                  L$NAME::
       002000      103                                                                 .ASCII /C/
       002001      132                                                                 .ASCII /Z/
       002002      125                                                                 .ASCII /U/
       002003      104                                                                 .ASCII /D/
       002004      103                                                                 .ASCII /C/
       002005      000                                                                 .BYTE   0
       002006      000                                                                 .BYTE   0
       002007      000                                                                 .BYTE   0
       002010                                                                  L$REV::
       002010      103                                                                 .ASCII  /C/
       002011                                                                  L$DEPO::
       002011      060                                                                 .ASCII  /0/
       002012                                                                  L$UNIT::
       002012  000001                                                                  .WORD   T$PTHV
       002014                                                                  L$TIML::
       002014  000000                                                                  .WORD   0
       002016                                                                  L$HPCP::
       002016  114566                                                                  .WORD   L$HARD
       002020                                                                  L$SPCP::
       002020  115032                                                                  .WORD   L$SOFT
       002022                                                                  L$HPTP::
       002022  064756                                                                  .WORD   L$HW
       002024                                                                  L$SPTP::
       002024  064774                                                                  .WORD   L$SW
       002026                                                                  L$LADP::
       002026  115616                                                                  .WORD   L$LAST
       002030                                                                  L$STA::
       002030  000000                                                                  .WORD   0
       002032                                                                  L$CO::
       002032  000000                                                                  .WORD   0
       002034                                                                  L$DTYP::
       002034  000001                                                                  .WORD   1
       002036                                                                  L$APT::
       002036  000000                                                                  .WORD   0
       002040                                                                  L$DTP::
       002040  064744                                                                  .WORD   L$DISPATCH
       002042                                                                  L$PRIO::
       002042  000340                                                                  .WORD   PRIO7
       002044                                                                  L$ENVI::
       002044  000000                                                                  .WORD   0
       002046                                                                  L$EXP1::
       002046  000000                                                                  .WORD   0
       002050                                                                  L$MREV::
       002050      003                                                                 .BYTE   C$REVISION
       002051      003                                                                 .BYTE   C$EDIT
```

```
002052
002052  000000                          L$EF::
002054  000000                                  .WORD   0
002056                                  .WORD   0
002056  000000                          L$SPC::
002060                                  .WORD   0
002060  065304                          L$DEVP::
002062                                  .WORD   L$DVTYP
002062  110006                          L$REPP::
002064                                  .WORD   L$RPT
002064  000000                          L$EXP4::
002066                                  .WORD   0
002066  000000                          L$EXP5::
002070                                  .WORD   0
002070  000000                          L$AUT::
002072                                  .WORD   0
002072  000000                          L$DUT::
002074                                  .WORD   0
002074  000000                          L$LUN::
002076                                  .WORD   0
002076  065330                          L$DESP::
002100                                  .WORD   L$DESC
002100  104035                          L$LOAD::
002102                                  EMT     E$LOAD
002102  065002                          L$ETP::
002104                                  .WORD   L$ERRTBL
002104  111036                          L$ICP::
002106                                  .WORD   L$INIT
002106  112522                          L$CCP::
002110                                  .WORD   L$CLEAN
002110  112520                          L$ACP::
002112                                  .WORD   L$AUTO
002112  111030                          L$PRT::
002114                                  .WORD   L$PROT
002114  000000                          L$TEST::
002116                                  .WORD   0
002116  000000                          L$DLY::
002120                                  .WORD   0
002120  000000                          L$HIME::
                                        .WORD   0
```

```
                        ;USEFUL INSTRUCTION DEFINITIONS

                        .MACRO  AND ARG,ADR                     ;LOGICAL AND INSTRUCTION
                                .LIST
                                                                        BIC #^C<ARG>,ADR
                                .NLIST
                        .ENDM

                        .MACRO  OR ARG,ADR                      ;LOGICAL OR INSTRUCTION
                                .LIST
                                                                        BIS #ARG,ADR
                                .NLIST
                        .ENDM

                        .MACRO  PUSH ARG                        ;PUSH INSTRUCTION
                                .IRP X,<ARG>
                                .LIST
                                                                        MOV X,-(SP)
                                .NLIST
                                .ENDM
                        .ENDM

                        .MACRO  POP ARG                         ;POP INSTRUCTION
                                .IRP X,<ARG>
                                .LIST
                                                                        MOV (SP)+,X
                                .NLIST
                                .ENDM
                        .ENDM

                        .MACRO  .BR ADR                                 ;A BRANCH TO THE NEXT LOCATION
                                .IF P2
                                    .IF NE .-ADR
                                        .ERROR ;ILLEGAL .BR TO ADR
                                    .ENDC
                                .ENDC
                        .ENDM

                        .MACRO  ASSUME FIRST CONDITION SECOND
                                    .IF CONDITION <FIRST>-<SECOND>
                                    .IFF
                                        .ERROR ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
                                    .ENDC
                        .ENDM
```

```
      1         062620                  STOSIZ = 26000.                 ;STORAGE SIZE
      2
      3                                 ;       THIS LOCATION MUST BE AT THIS POSITION. SEPERATE CODE, STORED IN
      4                                 ;       THE PAK FILE, WAS ASSEMBLED WITH THIS ADDRESS
      5                                 ;
      6 002122                                  ASSUME  . EQ 2122
      7
      8 002122                          STORAG: .BLKB   STOSIZ
      9
     10 064742                                  ASSUME  . LT 100000
```

```
                        .SBTTL  DISPATCH TABLE

                        ;++
                        ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
                        ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
                        ;--

                        DISPATCH 4
064742                                                                  .WORD   4
064742  000004                                                  L$DISPATCH::
064744                                                                  .WORD   T1
064744  112544                                                            .WORD   T2
064746  113720                                                            .WORD   T3
064750  114016                                                            .WORD   T4
064752  114054
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
                                ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
                                ;--

064754                                  BGNHW   DFPTBL
064754  000006                                                        .WORD   L10000-L$HW/2
064756                                                          L$HW::
064756                                                          DFPTBL::

064756  172150                  .WORD   172150          ; UNIBUS ADDRESS
064760  000154                  .WORD   154             ; VECTOR ADDRESS
064762  000005                  .WORD   5.              ; BR LEVEL
064764  000077                  .WORD   63.             ; UNIBUS BURST RATE
064766  000000                  .WORD   0.              ; LOGICAL DRIVE NUMBER
064770  000000                  .WORD   0.              ; CUSTOMER DATA AREA
064772                          ENDHW
064772                                                  L10000:
```

```
                                .SBTTL  SOFTWARE P-TABLE

                                ;++
                                ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                                ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                                ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                                ; AT RUN TIME.
                                ;--

                                        BGNSW   SFPTBL
                                                                                .WORD   L10001-L$SW/2
                                                                        L$SW::
                                                                        SFPTBL::

                                                                ;OFFSET         USE
                                .WORD   32.                     ; 0.    ERROR LIMIT
                                .WORD   0.                      ; 2.    DATA TRANSFER LIMIT (MEGABITS)
                                .WORD   ^B0100000100000000      ; 4.    SINGLE BIT QUESTIONS
                                ENDSW
                                                                        L10001:

                                ENDMOD
```

```
      1                                 .SBTTL  GLOBAL EQUATES SECTION
      2
      3 065002                                  BGNMOD
      4
      5                                 ;++
      6                                 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
      7                                 ; ARE USED IN MORE THAN ONE TEST.
      8                                 ;--
      9
     10 065002                                  EQUALS
                                       ;
                                       ; BIT DIFINITIONS
                                       ;
             100000                    BIT15== 100000
             040000                    BIT14== 40000
             020000                    BIT13== 20000
             010000                    BIT12== 10000
             004000                    BIT11== 4000
             002000                    BIT10== 2000
             001000                    BIT09== 1000
             000400                    BIT08== 400
             000200                    BIT07== 200
             000100                    BIT06== 100
             000040                    BIT05== 40
             000020                    BIT04== 20
             000010                    BIT03== 10
             000004                    BIT02== 4
             000002                    BIT01== 2
             000001                    BIT00== 1
                                       ;
             001000                    BIT9==  BIT09
             000400                    BIT8==  BIT08
             000200                    BIT7==  BIT07
             000100                    BIT6==  BIT06
             000040                    BIT5==  BIT05
             000020                    BIT4==  BIT04
             000010                    BIT3==  BIT03
             000004                    BIT2==  BIT02
             000002                    BIT1==  BIT01
             000001                    BIT0==  BIT00
                                       ;
                                       ; EVENT FLAG DEFINITIONS
                                       ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                       ;
             000040                    EF.START==      32.                     ; START COMMAND WAS ISSUED
             000037                    EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
             000036                    EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
             000035                    EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
             000034                    EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                                       ;
                                       ;
                                       ; PRIORITY LEVEL DEFINITIONS
                                       ;
             000340                    PRI07== 340
             000300                    PRI06== 300
             000240                    PRI05== 240
             000200                    PRI04== 200
```

GLOBAL EQUATES SECTION

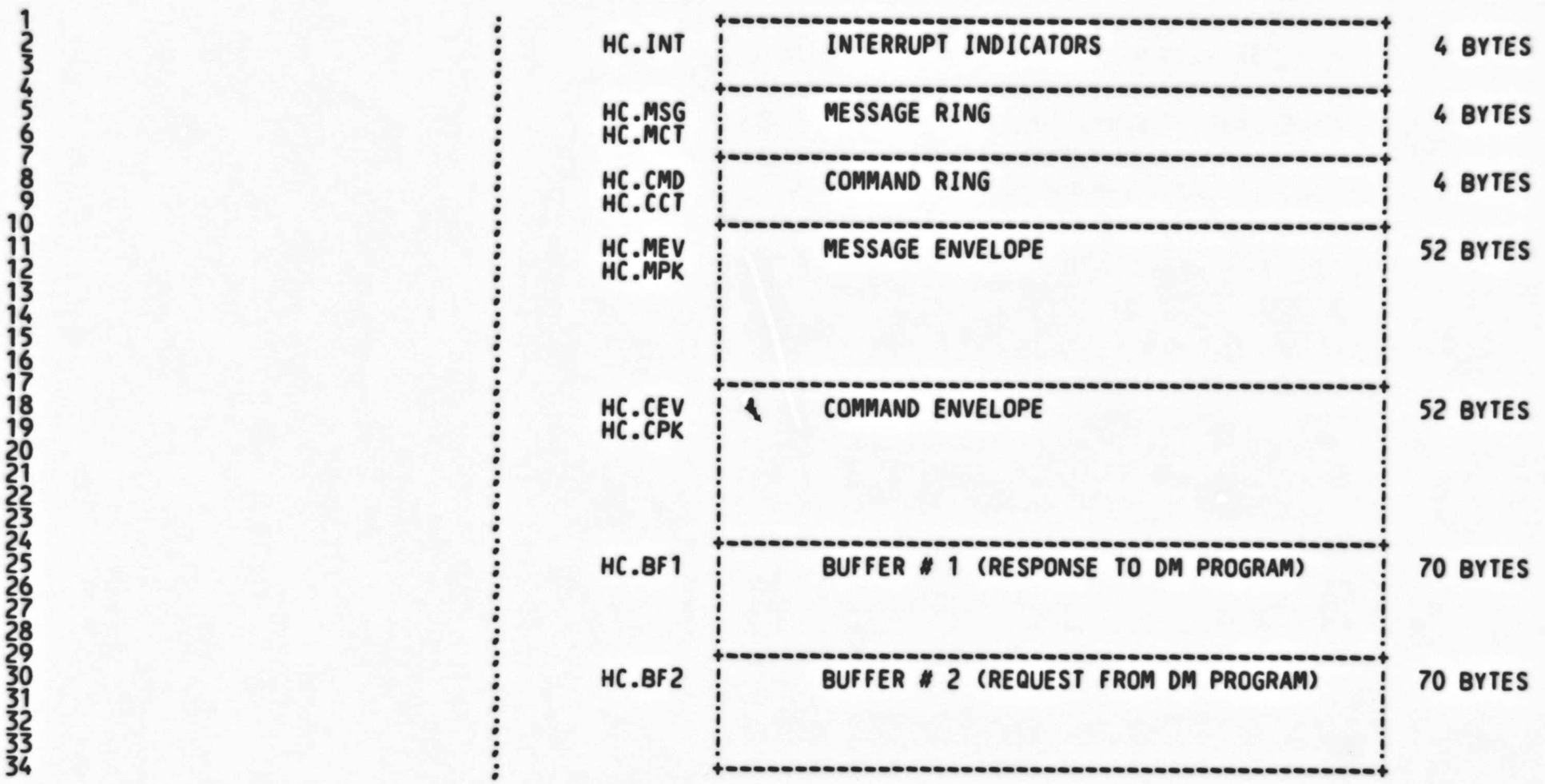
```
000140                  PRIO3== 140
000100                  PRIO2== 100
000040                  PRIO1== 40
000000                  PRIO0== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==       4
000010                  LOT==      10
000020                  ADR==      20
000040                  IDU==      40
000100                  ISR==     100
000200                  UAM==     200
000400                  BOE==     400
001000                  PNT==    1000
002000                  PRI==    2000
004000                  IXE==    4000
010000                  IBE==   10000
020000                  IER==   20000
040000                  LOE==   40000
100000                  HOE==  100000
11
12      000015          CR=     15                      ;VALUE TO PASS TO PRINT MACRO TO END LINE
```

```
                        ;MACRO DEFINITIONS FOR GLOBAL EQUATES

                        ;THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE
                        ;
                        ;CALLING SEQUENCE MUST BE
                        ;
                        ;       TABLE
                        ;       ITEM    NAME    BYTES
                        ;       ITEM    NAME    BYTES
                        ;       ITEM    NAME    BYTES
                        ;       END     SIZE
                        ;
                        ;TABLE DEFINES THAT A TABLE IS ABOUT TO BE DEFINED AND END TERMINATES THE DEFINITION.
                        ;ANY NUMBER OF ITEM LINES CAN APPEAR. NAME IS THE NAME OF THE SYMBOL BEING EQUATED TO
                        ;THE INDEX. THE INDEX ALWAYS STARTS AT ZERO. BYTES SPECIFIES THE SIZE OF THE VALUE TO BE
                        ;STORED AT THAT INDEX IN BYTES. THE SIZE ARGUMENT TO THE END STATEMENT IS OPTIONAL, IT
                        ;BE EQUATED TO THE SIZE OF THE TABLE IN BYTES. THE SYMBOL TINDEX IS USED TO KEEP TRACK
                        ;OF THE INDEX VALUE AND WILL BE EQUAL TO THE SIZE OF THE TABLE AFTER THE END STATEMENT.

                        .MACRO TABLE
                                TINDEX=0
                        .ENDM

                        .MACRO ITEM NAME BYTES
                                NAME=TINDEX
                                TINDEX=TINDEX+BYTES
                        .ENDM

                        .MACRO END SIZE
                                .IF NB SIZE
                                SIZE=TINDEX
                                .ENDC
                        .ENDM
```

```
                        ;UDA BIT DEFINITIONS
                        ;
                        ;UDASA 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
                        ;
                        ;UDASA REGISTER ERROR STATUS BITS
                        ;
003777                  SA.ERC= 003777          ;ERROR CODE
                        ;
                        ;UDASA REGISTER STEP ONE READ BITS
                        ;
002000                  SA.NV=  002000          ;NON SETTABLE INTERRUPT VECTOR
001000                  SA.A2=  001000          ;22 BIT ADDRESS BUS
000400                  SA.DI=  000400          ;ENHANCED DIAGNOSTICS
                        ;       000377          ;ALL BITS RESERVED
                        ;
                        ;UDASA REGISTER STEP ONE WRITE BITS
                        ;
000177                  SA.VEC= 000177          ;INTERRUPT VECTOR (DIVIDED BY 4)
000200                  SA.INT= 000200          ;INTERRUPT ENABLE DURING INITIALIZATION
003400                  SA.MSG= 003400          ;MESSAGE RING LENGTH
034000                  SA.CMD= 034000          ;COMMAND RING LENGTH
040000                  SA.WRP= 040000          ;WRAP BIT
100000                  SA.STP= 100000          ;STEP - MUST ALWAYS BE WRITTEN A ONE
                        ;
000400                  SA.MS1= 000400          ;LSB OF MESSAGE RING LENGTH
004000                  SA.CM1= 004000          ;LSB OF COMMAND RING LENGTH
                        ;
                        ;UDASA REGISTER STEP TWO READ BITS
                        ;
000007                  SA.MSE= 000007          ;MESSAGE RING LENGTH ECHO
000070                  SA.CME= 000070          ;COMMAND RING LENGTH ECHO
                        ;       000100          ;RESERVED
000200                  SA.STE= 000200          ;STEP ECHO
003400                  SA.CTP= 003400          ;CONTROLLER TYPE
                        ;
                        ;UDASA REGISTER STEP TWO WRITE BITS
                        ;
000001                  SA.PRG= 000001          ;ENABLE VAX UNIBUS ADAPTER PURGE INTERRUPT
                        ;       177776          ;LOW ORDER MESSAGE RING BYTE ADDRESS
```

```
                                ;UDASA REGISTER STEP THREE READ BITS
                                
000177                          SA.VCE= 000177          ;INTERRUPT VECTOR ECHO
000200                          SA.INE= 000200          ;INTERRUPT ENABLE ECHO
000400                          SA.NVE= 000400          ;VECTOR NOT PROGRAMMABLE
                                ;       003000          ;RESERVED

                                ;UDASA REGISTER STEP THREE WRITE BITS

                                ;       077777          ;HIGH ORDER MESSAGE RING BYTE ADDRESS
100000                          SA.TST= 100000          ;PURGE POLE TEST ENABLE

                                ;UDASA REGISTER STEP FOUR READ BITS

000017                          SA.MCV= 000017          ;UDA MICROCODE VERSION
000360                          SA.CNT= 000360          ;CONTROLLER TYPE
                                ;       003400          ;RESERVED

                                ;UDASA REGISTER STEP FOUR WRITE BITS

000001                          SA.GO=  000001          ;GO BIT TO START UDA FIRMWARE
000002                          SA.LFC= 000002          ;LAST FAILURE CODE REQUEST
000374                          SA.BST= 000374          ;BURST LEVEL
```

```
                        ;COMMAND/MESSAGE DESCRIPTOR BIT DEFINITIONS

100000                  RG.OWN= 100000                  ;SET WHEN UDA OWNS RING
040000                  RG.FLG= 040000                  ;FLAG BIT

                        ;OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
                        ;AND TWO PACKET AND BUFFER AREAS.

000004                  HC.ISZ=  4.                     ;SIZE OF INTERRUPT INDICATOR WORDS
000004                  HC.RSZ=  4.                     ;SIZE OF RING IN BYTES
000004                  HC.ESZ=  4.                     ;SIZE OF ENVELOPE WORDS BEFORE PACKET
000060                  HC.PSZ= 48.                     ;SIZE OF COMMAND AND MESSAGE PACKETS
000106                  HC.BSZ= 70.                     ;SIZE OF BUFFER

000000                  HC.INT=  0.                     ;INTERRUPT INDICATOR WORDS START
000004                  HC.MSG= HC.INT+HC.ISZ           ;MESSAGE RING START
000006                  HC.MCT= HC.MSG+2.               ;MESSAGE RING CONTROL WORD
000010                  HC.CMD= HC.MSG+HC.RSZ           ;COMMAND RING START
000012                  HC.CCT= HC.CMD+2.               ;COMMAND RING CONTROL WORDS
000014                  HC.MEV= HC.CMD+HC.RSZ           ;MESSAGE ENVELOPE START
000020                  HC.MPK= HC.MEV+HC.ESZ           ;MESSAGE PACKET START
000100                  HC.CEV= HC.MPK+HC.PSZ           ;COMMAND ENVELOPE START
000104                  HC.CPK= HC.CEV+HC.ESZ           ;COMMAND PACKET START
000164                  HC.BF1= HC.CPK+HC.PSZ           ;FIRST BUFFER
000272                  HC.BF2= HC.BF1+HC.BSZ           ;SECOND BUFFER

000400                  HC.SIZ= HC.BF2+HC.BSZ           ;TOTAL SIZE OF HOST COMM AREA

                        ;VIRTUAL CIRCUIT IDENTIFIERS

000000                  MSCP=    0                      ;MSCP CIRCUIT
000001                  LOG=     1                      ;LOG CIRCUIT
177777                  DIAG=   -1                      ;DIAGNOSTIC CIRCUIT
001000                  DUP=    1000                    ;DIAGNOSTIC AND UTILITIES PROTOCOL
```


HC.INT
INTERRUPT INDICATORS
4 BYTES
HC.MSG
HC.MCT
MESSAGE RING
4 BYTES
HC.CMD
HC.CCT
COMMAND RING
4 BYTES
HC.MEV
HC.MPK
MESSAGE ENVELOPE
52 BYTES
HC.CEV
HC.CPK
COMMAND ENVELOPE
52 BYTES
HC.BF1
BUFFER # 1 (RESPONSE TO DM PROGRAM)
70 BYTES
HC.BF2
BUFFER # 2 (REQUEST FROM DM PROGRAM)
70 BYTES

```
                        ;COMMAND PACKET OPCODES
                        
000001          OP.ABO=   1                     ;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
000022          OP.ERS=  22                     ;ERASE COMMAND
000023          OP.FLU=  23                     ;FLUSH COMMAND
000002          OP.GCS=   2                     ;GET COMMAND STATUS COMMAND
000003          OP.GUS=   3                     ;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=   4                     ;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
000007          OP.SEX=   7                     ;SERIOUS EXCEPTION END PACKET
000100          OP.AVA= 100                     ;AVAILABLE ATTENTION MESSAGE
000101          OP.DUP= 101                     ;DUPLICATE UNIT NUMBER ATTENTION MESSAGE
000102          OP.SHC= 102                     ;SHADOW COPY COMPLETE ATTENION MESSAGE
000103          OP.RLC= 103                     ;RESET COMMAND LIMIT ATTENTION MESSAGE

000001          OP.GSS=   1                     ;DUP GET DUST STATUS
000002          OP.ESP=   2                     ;DUP EXECUTE SUPPLIED PROGRAM
000003          OP.ELP=   3                     ;DUP EXECUTE LOCAL PROGRAM
000004          OP.SSD=   4                     ;DUP SEND DUST DATA
000005          OP.RSD=   5                     ;DUP RECEIVE DUST DATA

                        ;NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING THE END
                        ;PACKET FLAG TO THE COMMAND OPCODE. FOE EXAMPLE, A READ COMMAND'S END PACKET
                        ;CONTAINS THE VALUE OP.RD+OP.END IN ITS OPCODE FIELD. THE INVALID COMMAND END
                        ;PACKET CONTAINS JUST THE END PACKET FLAG (I.E., OP.END) IN ITS OPCODE FIELD.
                        ;THE SERIOUS EXCEPTION END PACKET CONTAINS THE SUM OF THE END PACKET FLAG
                        ;PLUS THE SERIOUS EXCEPTION OPCODE SHOWN ABOVE (I.E., OP.SEX+OP.END) IN ITS
                        ;OPCODE FIELD.
                        ;
                        ;COMMAND OPCODE BITS 3 THROUGH 5 INDICATE THE COMMAND CLASS, WHICH IS ENCODED
                        ;AS FOLLOWS:
                        ;    000  IMMEDIATE COMMANDS
                        ;    001  SEQUENTIAL COMMANDS
                        ;    010  NON-SEQUENTIAL COMMANDS THAT DO NOT INCLUDE A BUFFER DESCRIPTOR
                        ;    100  NON-SEQUENTIAL COMMANDS THAT DO INCLUDE A BUFFER DESCRIPTOR
```

```
                        ;COMMAND MODIFIERS
                        ;
                        ;       = 020000                ;CLEAR SERIOUS EXCEPTION
040000                  MD.CMP= 040000                  ;COMPARE
100000                  MD.EXP= 100000                  ;EXPRESS REQUEST
010000                  MD.ERR= 010000                  ;FORCE ERROR
004000                  MD.SCH= 004000                  ;SUPPRESS CACHING (HIGH SPEED)
002000                  MD.SCL= 002000                  ;SUPPRESS CACHING (LOW SPEED)
000100                  MD.SEC= 000100                  ;SUPPRESS ERROR CORRECTION
000400                  MD.SER= 000400                  ;SUPPRESS ERROR RECOVERY
000200                  MD.SSH= 000200                  ;SUPPRESS SHADOWING
000100                  MD.WBN= 000100                  ;WRITE-BACK (NON-VOLATILE)
000400                  MD.WBV= 000400                  ;WRITE BACK (VOLATILE)
000020                  MD.SEQ= 000020                  ;WRITE SHADOW SET ONE UNIT AT A TIME
000001                  MD.SPD= 000001                  ;SPIN-DOWN
000001                  MD.FEU= 000001                  ;FLUSH ENTIRE UNIT
000002                  MD.VOL= 000002                  ;VOLATILE ONLY
000001                  MD.NXU= 000001                  ;NEXT UNIT
000001                  MD.RIP= 000001                  ;ALLOW SELF DESTRUCTION
000002                  MD.IMF= 000002                  ;IGNORE MEDIA FORMAT ERROR
000004                  MD.SWP= 000004                  ;SET WRITE PROTECT
000010                  MD.CWB= 000010                  ;CLEAR WRITE-BACK DATA LOST
000001                  MD.PRI= 000001                  ;PRIMARY REPLACEMENT BLOCK

                        ;END PACKET FLAGS

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

                        ;CONTROLLER FLAGS

000200                  CF.ATN= 000200                  ;ENABLE 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
```

```
                        ;UNIT FLAGS
                        ;
000001                  UF.CMR= 000001                  ;COMPARE READS
000002                  UF.CMW= 000002                  ;COMPARE WRITES
100000                  UF.RPL= 100000                  ;HOST INITIATED BAD BLOCK REPLACEMENT
040000                  UF.INA= 040000                  ;INACTIVE SHADOW SET UNIT
004000                  UF.SCH= 004000                  ;SUPPRESS CACHING (HIGH SPEED)
002000                  UF.SCL= 002000                  ;SUPPRESS CACHING (LOW SPEED)
000100                  UF.WBN= 000100                  ;WRITE-BACK (NON-VOLATILE)
020000                  UF.WPH= 020000                  ;WRITE PROTECT (HARDWARE)
001000                  UF.WPS= 001000                  ;WRITE PROTECT (SOFTWARE OR VOLUME)
000004                  UF.576= 000004                  ;576 BYTE SECTORS
                        ;
                        ;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.UADR= 16.                     ;UNIBUS ADDRESS OF BUFFER DESCRIPTOR
000034                  P.LBN=  28.                     ;LOGICAL BLOCK NUMBER
                        ;
                        ;               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 / RESERVED
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
                        ;
                        ;               EXECUTE SUPPLIED PROGRAM COMMAND PACKET OFFSETS:
000024                  P.DMDT= 20.                     ;DMDT TERMINAL ADDRESS (MAINT WRITE ONLY)
000034                  P.OVRL= 28.                     ;BUFFER DESCRIPTOR FOR OPERLAYS
```

```
                        ;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.                     ;STATUS
000014                  P.BCNT= 12.                     ;BYTE COUNT
000034                  P.FBBK= 28.                     ;FIRST BAD BLOCK

                        ;               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
000034                  P.MEDI= 28.                     ;MEDIA TYPE 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.                     ;RBNS / TRACK
000057                  P.RCTC= 47.                     ;RCT COPIES

                        ;               ONLINE AND SET UNIT CHARACTERISTICS END PACKET AND AVAILABLE
                        ;               ATTENTION MESSAGE 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
000034                  P.MEDI= 28.                     ;MEDIA TYPE IDENTIFIER
000040                  P.SHUN= 32.                     ;SHADOW UNIT
000042                  P.SHST= 34.                     ;SHADOW STATUS
000044                  P.UNCL= 36.                     ;UNIT COMMAND LIMIT
000050                  P.UNSZ= 40.                     ;UNIT SIZE
000054                  P.VSER= 44.                     ;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

                        ;               GET DUST STATUS END PACKET OFFSETS:
000014                  P.DEXT= 12.                     ;EXTENSION FOR DOWNLINE LOADABLE PROGRAM
000017                  P.DFLG= 15.                     ;FLAGS
000020                  P.DPRG= 16.                     ;PROGRESS INDICATOR FOR REMOTE PROGRAM
000024                  P.DTMO= 20.                     ;TIMEOUT
```

GLOBAL EQUATES SECTION

```
                        ;STATUS AND EVENT CODE 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 FORMAT 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
                        
                        ;DUP MESSAGE TYPES
                        ;
010000                  DU.QUE = 10000                  ;QUESTION
020000                  DU.DFL = 20000                  ;DEFAULT QUESTION
030000                  DU.INF = 30000                  ;INFORMATION
040000                  DU.TER = 40000                  ;TERMINATOR
050000                  DU.FTL = 50000                  ;FATAL ERROR
060000                  DU.SPC = 60000                  ;SPECIAL
```

```
                                ;CONTROLLER TABLE DEFINITIONS
                                ;
                                ;ONE TABLE WILL BE SET UP BY INITIALIZE SECTION FOR EACH UDA SELECTED
                                ;FOR TESTING.  TABLES ARE CONTIGUOUS.  THE END OF THE TABLES IS
                                ;MARKED BY A WORD OF ZEROS.
                                ;
                                ;THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF CTABS.
                                ;THE NUMBER OF TABLES IS CONTAINED IN CTRLRS.

065002                          TABLE                                   ;START A TABLE DEFINITION

065002                          ITEM C.UADR     2                       ;UNIBUS ADDRESS OF UDAIP REGISTER
065002                          ITEM C.UNIT     2
        000077                          CT.UNT= 000077                  ; LOGICAL UNIT NUMBER (FIRST)
        100000                          CT.AVL= BIT15                   ; SET WHEN NOT AVAILABLE FOR TESTING
065002                          ITEM C.VEC      2
        000777                          CT.VEC= 000777                  ; VECTOR ADDRESS
        007000                          CT.BRL= 007000                  ; BR LEVEL
065002                          ITEM C.BST      2                       ; BURST LEVEL
065002                          ITEM C.JSR      2                       ;INTERRUPT SERVICE ROUTINE FOR CONTROLLER
065002                          ITEM C.JAD      2                       ;  THESE TWO WORDS LOADED WITH [JSR R0,UDASRV]
065002                          ITEM C.FLG      2                       ;FLAGS
        000002                          CT.RN=  BIT1                    ;DM PROGRAM RUNNING
        000004                          CT.CMD= BIT2                    ;COMMAND ISSUED, WAITING FOR RESPONSE
        000010                          CT.MSG= BIT3                    ;MESSAGE RESPONSE RECEIVED
                                                                        ;WHENEVER THIS BIT IS SET, CT.CMD IS CLEARED
        000020                          CT.REQ= BIT4                    ;BUFFER HAS BEEN GIVEN TO UDA FOR REQUEST
                                                                        ;SET WHENEVER READ STUD DATA COMMAND
                                                                        ;GIVEN TO UDA
        000040                          CT.U50= BIT5                    ;CONTROLLER IS UDA50 IF SET/UDA52 IF CLEARED
065002                          ITEM C.RING     2                       ;RING BUFFER ADDRESS
065002                          ITEM C.DR0      2                       ;POINTER TO DRIVE TABLES
065002                          ITEM C.DR1      2                       ;  IF ZERO, NO DRIVE TABLE EXISTS
065002                          ITEM C.DR2      2
065002                          ITEM C.DR3      2
065002                          ITEM C.DR4      2
065002                          ITEM C.DR5      2
065002                          ITEM C.DR6      2
065002                          ITEM C.DR7      2
065002                          ITEM C.TO       2                       ;TIMEOUT COUNTER
065002                          ITEM C.TOH      2                       ;  (TWO WORDS)
065002                          ITEM C.REF      2                       ;COMMAND REFERENCE NUMBER

065002                          END C.SIZE                              ;SIZE OF CONTROLLER TABLE IN BYTES
```

```
                                ;DRIVE TABLE DEFINITIONS
                                ;
                                ;ONE DRIVE TABLE WILL BE SET UP BY THE INITIALIZE SECTION FOR EACH
                                ;DRIVE SELECTED FOR TESTING.  EACH TABLE IS POINTED TO BY A
                                ;WORD IN THE CONTROLLER TABLE ON WHICH THE DRIVE EXISTS.

065002                          TABLE                                   ;START A TABLE DEFINITION

065002                          ITEM D.DRV      2                       ;DRIVE NUMBER
065002                          ITEM D.UNIT     2
000077                                  DT.UNT= 000077                  ; LOGICAL UNIT NUMBER OF DRIVE
100000                                  DT.AVL= BIT15                   ; SET WHEN NOT AVAILABLE FOR TESTING
065002                          ITEM D.PRM      2                       ;HARDWARE QUESTION FLAGS
040000                                  D.IW    =BIT14                  ;INITIAL WRITE
020000                                  D.DCY   =BIT13                  ;DIAGNOSTIC CYLINDERS
010000                                  D.ECC   =BIT12                  ;ECC CORRECTION ENABLED
004000                                  D.RO    =BIT11                  ;READ ONLY
002000                                  D.WO    =BIT10                  ;WRITE ONLY
001000                                  D.RET   =BIT9                   ;RETRIES ENABLED
000400                                  D.CYL   =BIT8                   ;START/END CYLINDERS SPECIFIED
000100                                  D.SEQ   =BIT6                   ;SEQUENTIAL ACCESS
000040                                  D.BE    =BIT5                   ;BEGIN/END BLOCKS USED
000020                                  D.TR    =BIT4                   ;WHEN D.BE=0: 1 - TRACKS, 0 - GROUPS
000010                                  D.WC    =BIT3                   ;WRITE CHECKS ENABLED
000004                                  D.WCA   =BIT2                   ;ALWAYS WRITE CHECK
000002                                  D.DC    =BIT1                   ;DATA COMPARES ENABLED
000001                                  D.DCA   =BIT0                   ;ALWAYS DATA COMPARE
011012                                          DDEF=D.ECC+D.WC+D.DC+D.RET      ;DEFAULT D.PRM
140200                                          D.ZERO=BIT15+BIT7+D.IW  ;BITS TO BE CLEARED
065002                  ITEM D.PAT      2                               ;DATA PATTERN NUMBER
065002                  ITEM D.BB       2                               ;BAD BLOCK COUNT
065002                  ITEM D.BB01     4                               ;BAD BLOCK 1
065002                  ITEM D.BB02     4                               ;         2
065002                  ITEM D.BB03     4                               ;         3
065002                  ITEM D.BB04     4                               ;         4
065002                  ITEM D.BB05     4                               ;         5
065002                  ITEM D.BB06     4                               ;         6
065002                  ITEM D.BB07     4                               ;         7
065002                  ITEM D.BB08     4                               ;         8
065002                  ITEM D.BB09     4                               ;         9
065002                  ITEM D.BB10     4                               ;        10
065002                  ITEM D.BB11     4                               ;        11
065002                  ITEM D.BB12     4                               ;        12
065002                  ITEM D.BB13     4                               ;        13
065002                  ITEM D.BB14     4                               ;        14
065002                  ITEM D.BB15     4                               ;        15
065002                  ITEM D.BB16     4                               ;        16
```

```
      1 065002                          ITEM D.BEC      2               ;BEGIN/END SET COUNT
      2 065002                          ITEM D.BGN1     4               ;BEGIN BLOCK 1
      3 065002                          ITEM D.END1     4               ;END
      4 065002                          ITEM D.BGN2     4               ;BEGIN BLOCK 2
      5 065002                          ITEM D.END2     4               ;END
      6 065002                          ITEM D.BGN3     4               ;BEGIN BLOCK 3
      7 065002                          ITEM D.END3     4               ;END
      8 065002                          ITEM D.BGN4     4               ;BEGIN BLOCK 4
      9 065002                          ITEM D.END4     4               ;END
     10 065002                          ITEM D.BCYL     4               ;BEGIN CYLINDER
     11 065002                          ITEM D.ECYL     4               ;END CYLINDER
     12 065002                          ITEM D.XFRW     2               ;MEGABITS WRITTEN COUNT
     13 065002                          ITEM D.XFRR     2               ;MEGABITS READ COUNT
     14 065002                          ITEM D.HERR     2               ;HARD ERROR COUNTER
     15 065002                          ITEM D.SERR     2               ;SOFT ERROR COUNTER
     16 065002                          ITEM D.SEEK     2               ;NUMBER OF SEEKS X1000
     17 065002                          ITEM D.ECCC     2               ;ECC COUNTER
     18 065002                          ITEM D.SERN     6               ;DRIVE SERIAL NUMBER
     23
     24 065002                          END D.SIZE                      ;SIZE OF DRIVE TABLE IN BYTES
     25
     26                                 ;DM PROGRAM HEADER DEFINITIONS
     27
     28          000000                 DMTRLN= 0                       ;OFFSET TO SIZE OF PROGRAM NEEDING DOWNLINE LOAD
     29          000004                 DMOVRL= 4                       ;OFFSET TO SIZE OF OVERLAY
     30          000040                 DMMAIN= 40                      ;OFFSET TO FIRST WORD OF MAIN PROGRAM
     31          001000                 DMFRST= 1000                    ;ADDRESS IN DM FILE CONTAINING FIRST BYTE OF HEADER
```

```
                        ;PRINT CHARACTER
                        ;  ARGUMENT MUST BE SOURCE STATEMENT TO MOVE CHARACTER TO PRINT (MOV ARG,R0)
                        ;       EX: "PRINT R1" WILL PRINT THE CHARACTER IN R1
                        ;  SPECIAL CASE: "PRINT #CR" WILL PRINT END OF LINE SEQUENCE
                        ;  THE PRINTING IS DONE AT THE MODE OF THE LAST PRINT LINE CALL
                        ;       IE., PNTX, PNTB, PNTX, PNTS

                        .MACRO PRINT ARG1
                                .IF DIF <ARG1>,R0
                                        .LIST
                                                                                MOVB ARG1,R0
                                        .NLIST
                                .ENDC
                                .LIST
                                                                                CALL CPNT
                                .NLIST
                        .ENDM

                        ;PROCESSING MACRO FOR NEXT SET OF FORMATTED MESSAGE MACROS

                        .MACRO PNT... RTN,ADR,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                PNT.CT=0
                                .IRP AA,<ARG8,ARG7,ARG6,ARG5,ARG4,ARG3,ARG2,ARG1>
                                        .IF NB,<AA>
                                                .LIST
                                                                                MOV AA,-(SP)
                                                .NLIST
                                                PNT.CT=PNT.CT+2
                                        .ENDC
                                .ENDM
                                .LIST
                                                                                JSR R1,RTN
                                                                                .WORD ADR
                                                                                .WORD PNT.CT
                                .NLIST
                        .ENDM

                        ;PRINT FORMATTED MESSAGE MACROS
                        ;  USE THESE MACROS TO PRINT A FORMATTED MESSAGE
                        ;  FIRST ARGUMENT MUST BE ADDRESS OF FIRST CHARACTER OF MESSAGE STRING
                        ;       TO BE PUT INTO WORD (.WORD ARG)
                        ;  UP TO 8 SOURCE STATEMENTS MAY FOLLOW TO SPECIFY PARAMETERS TO BE
                        ;  USED BY THE FORMAT

                        .MACRO PNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                PNT... LPNTF ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                        .ENDM
                        .MACRO PNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                PNT... LPNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                        .ENDM
                        .MACRO PNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                PNT... LPNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                        .ENDM
                        .MACRO PNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                PNT... LPNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                        .ENDM
                        .MACRO PNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
```

GLOBAL EQUATES SECTION

```
                                        PNT... LPNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                .ENDM
```

```
                                .SBTTL  GLOBAL DATA SECTION

                                ;++
                                ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
                                ; IN MORE THAN ONE TEST.
                                ;--

                                        ERRTBL
065002                                                                  L$ERRTBL::
065002  000000                  ERRTYP::      .WORD   0
065004  000000                  ERRNBR::      .WORD   0
065006  000000                  ERRMSG::      .WORD   0
065010  000000                  ERRBLK::      .WORD   0

065012  104070                  PTYPE:  .WORD PF                      ;PRINT TYPE
065014     000      000         ERRCHR: .BYTE 0,0                     ;FIRST BYTE LOADED WITH OUTPUT CHARACTER
                                                                      ;SECOND BYTE REMAINS ZERO TO STOP OUTPUT
065016                          FFREE:: .BLKW 1                       ;FIRST FREE WORD IN MEMORY
065020                          FSIZE:: .BLKW 1                       ;SIZE OF FREE MEMORY IN WORDS
065022                          FMEM:   .BLKW 1                       ;COPY OF FFREE AT END OF INIT SECTION
065024                          FMEMS:  .BLKW 1                       ;COPY OF FSIZE AT END OF INIT SECTION
065026                          CTABS:: .BLKW 1                       ;START OF CONTROLLER TABLE STORAGE
065030                          CTRLRS: .BLKW 1                       ;COUNT OF UDA CONTROLLERS IN PTABLES
065032                          TSTTAB: .BLKW 1                       ;POINTER TO FIRST CONTROLLER TABLE UNDER TEST
065034                          DMPROG: .BLKW 1                       ;START ADDRESS OF UDA52 DM PROGRAM
065036                          D50PRG: .BLKW 1                       ;START ADDRESS OF UDA50 DM PROGRAM

065040                          KTBASA: .BLKW 1                       ;HIGH TWO BYTES OF BASE ADDRESS FOR KT ACCESS
065042                          KTBASO: .BLKW 1                       ;LOW BYTE OF ADDRESS FOR KT ACCESS

065044                          IFLAGS::.BLKW 1                       ;FLAGS FROM INIT CODE FOR TEST 4
          000002                                ICONT ==BIT1            ; CONTINUE EVENT FLAG
          000004                                IREST ==BIT2            ; RESTART FLAG
          000010                                ISTRT ==BIT3            ; START FLAG
          000020                                ISTRTH==BIT4            ; START FLAG HOLD FOR T4UPRM ROUTINE
065046  000000                  FNUM:     .WORD 0                     ;FILE # IN PAK FILE THAT IS CURRENTLY LOADED
065050  000000                  TNUM:     .WORD 0                     ;NUMBER OF TEST EXECUTING
065052                          URUN:     .BLKW 1                     ;NUMBER OF UNITS TO RUN AT ONE TIME
065054                          URNING:   .BLKW 1                     ;NUMBER OF UNITS STILL RUNNING
065056                          UCNT:     .BLKW 1                     ;COUNTER OF UNITS UNDER TEST
065060                          INTRCV:   .BLKW 1                     ;INTERRUPT RECEIVED FLAG FOR INT TESTING
```

GLOBAL DATA SECTION

```
                                FNAME:
                                        .ASCIZ\ZUDDC0.PAK\              ;NAME OF DATA FILE
                                        .EVEN
                                FDATA:  .WORD 0
                                FILOPN: .WORD 0                         ;FILE OPEN WHEN NON-ZERO
                                TEMP:   .BLKW 12.                       ;TEMPORY STORAGE FOR GMANI RESPONSES

                                U52EXT: .ASCII  "U52"
                                        .EVEN

                                TYPCNT: .WORD   0                       ; TYPE OF CONTROLLER WORD
                                TY.U50 = BIT1
                                TY.U52 = BIT0

                                IPADRS: .WORD   0                       ; EIGHT ENTRIES
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0

                                PAT16C: .WORD 1                         ;COUNT OF WORDS IN DATA PATTERN 16
                                PAT16W: .WORD 0                         ;WORD SEQUENCE FOR DATA PATTERN 16
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
                                        .WORD 0
```

```
                                ;CLOCK CONTROL
                                
065222  000000                  KW.CSR: .WORD 0                 ;CSR OF CLOCK
065224                          KW.BRL: .BLKW 1                 ;BR LEVEL
065226                          KW.VEC: .BLKW 1                 ;VECTOR
065230                          KW.HZ:  .BLKW 1                 ;HERTZ (50. OR 60.)
065232                          KW.EL:  .BLKW 2                 ;ELAPSED TIME
065236                          STIME:  .BLKW 2                 ;STATISTICAL REPORT TIMER

065242                          NXMAD:  .BLKW 1                 ;SET TO ALL ONES BY NON-EXISTANT ADDRESS
065244  177777                  KTMEM:  .WORD -1                ;SET TO ALL ONES IF NO KT EXISTS

065246                          T2WRR:  .BLKW 1                 ;WRITE/READ REGION
065250                          T2WRO:  .BLKW 1                 ;WRITE/READ OFFSET
065252                          T2DR:   .BLKW 1                 ;DIAGNOSE REGION


                                ;ERROR LOG CONTROL WORDS

065254                          LBUFS:  .BLKW 1                 ;START ADDRESS OF LOG/ZERO IF NONE
065256                          LBUFN:  .BLKW 1                 ;ADDRESS FOR MORE DATA FOR LOG
065260                          LBUFE:  .BLKW 1                 ;LAST ADDRESS AVAILABLE FOR LOG DATA

                                ;DISK DIAGNOSTIC DLL CONTROL WORDS

065262                          DLL:    .BLKW 1                 ;DOWNLINE LOAD RESPONSE CODE = 0 - NO DATA,
                                                                ;1 - PROGRAM PROVIDED, 2- PROGRAM NOT FOUND
065264                          DLLDR:  .BLKW 1                 ;DRIVE NUMBER REQUESTING PROGRAM
065266                          DLLV:   .BLKW 1                 ;A VALUE FROM DM PROGRAM TO BE RETURNED
065270                          DLLR:   .BLKW 1                 ;REGION
065272                          DLLADR: .BLKW 2                 ;ADDRESS WHERE PROGRAM STORED
065276                          DLLSIZ: .BLKW 1                 ;SIZE OF PROGRAM IN BYTES
065300                          DLLNAM: .BLKW 2                 ;NAME OF PROGRAM IN RAD50
```

```
                        .SBTTL  GLOBAL TEXT SECTION

                        ;++
                        ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                        ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                        ; MORE THAN ONE TEST.
                        ;--

                        ;
                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;
 065304                         DEVTYP <LOGICAL DISK DRIVE>
 065304                                                         L$DVTYP::
 065304  114  117  107                                                  .ASCIZ  /LOGICAL DISK DRIVE/
                                                                        .EVEN

                        ; TEST DESCRIPTION
                        ;
 065330                         DESCRIPT <CZUDCCO UDA & DISK DRV DIAG>
 065330                                                         L$DESC::
 065330  103  132  125                                                  .ASCIZ  /CZUDCCO UDA & DISK
                                                                        .EVEN
```

```
                                ;UNFORMATTED MESSAGES
                        
 065364     040     040     000  T4OPT7: .ASCIZ\  \
 065367     101     122     105  INITWC: .ASCIZ\ARE YOU SURE CUSTOMER DATA CAN BE DESTROYED\
```

```
                                ; FORMAT STATEMENTS USED IN PRINT CALLS
                                
 065443     045     124     000  ERRONE: .ASCIZ\%T\
 065446     045     116     000  ERRNL:  .ASCIZ\%N\
 065451     042     040     040  RNTIM:  .ASCIZ\"  RUNTIME '"D16":'"\
 065474     104     071     042  RNTIM1: .ASCIZ\D9":'"\
 065502     104     071     000  RNTIM2: .ASCIZ\D9\
 065505     042     040     040  ERRME1: .ASCIZ\"  *  *  *  ERROR PROCESSING MESSAGE STRING  *  *  *'N\
 065574     116     042     122  MXFERP: .ASCIZ\N'REACHED TRANSFER LIMIT - TESTING STOPPED'N\
 065651     116     042     125  ERRLIM: .ASCIZ\N'UNIT 'D6" REACHED ERROR LIMIT - WILL NO LONGER BE TESTED'N\
 065746     116     042     124  INTSTO: .ASCIZ\N'TESTING INTERRUPT ABILITY OF UDA AT ADR "D16" VEC "D9"..."\
 066043     042     103     117  INTST1: .ASCIZ\"COMPLETED'N\
 066060     116     042     103  INITWA: .ASCIZ\N'CUSTOMER DATA WILL BE DESTROYED ON:'NS5'UNIT'S5'UDA AT'S3'DRIVE'N\
 066164     045     123     066  INITWB: .ASCIZ\%S6%D2%S6%O6%S4%D3%N\
 066211     116     042     115  T4WARN: .ASCIZ\N'MANUAL INTERVENTION NOT ALLOWED.  TEST 4 USING DEFAULT PARAMETERS'N\
 066317     116     042     125  MESSG:  .ASCIZ\N'UNIT 'D6" UDA AT "D16" DRIVE "D9S\
 066363     116     042     115  T2WARN: .ASCIZ\N'MANUAL INTERVENTION NOT ALLOWED. TEST 2 RUNNING UNATTENDED'N\
 066462     116     042     124  T2CMS1: .ASCII\N'TEST #2 MANUAL INTERVENTION ON UNIT 'D8" UDA AT "D16" DRIVE "D9N\
 066564     042     124     117          .ASCII\"TO WRITE AND READ MEMORY:'N\
 066620     042     040     040          .ASCII\"  W DATA REGION OFFSET'N\
 066651     042     040     040          .ASCII\"  R REGION OFFSET'N\
 066675     042     124     117          .ASCII\"TO RUN A DIAGNOSTIC:'N\
 066724     042     040     040          .ASCII\"  D REGION'N\
 066741     042     124     117          .ASCII\"TO EXIT QUESTIONING:'N\
 066770     042     040     040          .ASCII\"  E'N\
 066776     042     104     101          .ASCIZ\'DATA, REGION AND OFFSET ARE HEX VALUES.'N\
 067051     042     077     040  T2CMS5: .ASCIZ\"? INPUT ERROR'N\
 067072     042     116     117  NOCLOCK:.ASCIZ\'NO LINE CLOCK AVAILABLE FOR TIMING EVENTS'N\
 067147     116     042     103  LOGM1:  .ASCIZ\N'CONTENTS OF ERROR LOG:"\
 067201     116     042     105  LOGM2:  .ASCIZ\N'END OF ERROR LOG'N\
 067226     116     042     105  LOGM3:  .ASCIZ\N'ERROR LOG IS EMPTY'N\

 067255     042     110     117  BASN0:  .ASCIZ\'HOST PROGRAM"\
 067274     042     125     116  BASN1:  .ASCIZ\'UNIBUS ADDRESSING"\
 067320     042     104     111  BASN2:  .ASCIZ\'DISK RESIDENT"\
 067340     042     104     111  BASN3:  .ASCIZ\'DISK FUNCTION"\
 067360     042     104     111  BASN4:  .ASCIZ\'DISK EXERCISER"\
 067401     042     040     040  BASL1:  .ASCIZ\"  DM PC:"D12\
 067417     042     040     040  BASL2:  .ASCIZ\"  UDA AT "D16\
 067436     042     040     040  BASL3:  .ASCIZ\"  DRIVE "D9\
 067453     000                  BAS:    .BYTE 0                         ;NULL TO PRINT NOTHING

 067454     122     066     122  BASLN:  .ASCIZ\R6R6R6R6\                ;USED TO PRINT BASIC LINE OF ERROR MESSAGE
```

```
      1 067465                          X1A:
      2 067465                          X2A:
      3 067465                          X3A:
      4 067465  042     111     040     X8A:    .ASCIZ\"I DON'T LIKE THE ANSWERS YOU GAVE TO THE HARDWARE QUESTIONS'N\
      5 067564  122     065     122     X1:     .ASCIZ\R5R6'UDA HAS MORE THAN ONE VECTOR, BR LEVEL OR BURST RATE'N\
      6 067660  122     065     122     X2:     .ASCIZ\R5R6'TWO UNITS SELECT THE SAME DRIVE'N\
      7 067727  122     065     122     X3:     .ASCIZ\R5R6'MORE THAN EIGHT DRIVES SELECTED ON THIS UDA'N\
      8 070012  122     064     042     X4:     .ASCII\R4'NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED'N\
      9 070103  042     120     114             .ASCIZ\"PLEASE START PROGRAM OVER AND TEST FEWER UNITS AT A TIME'N\
     10 070177  122     064     042     X6:     .ASCIZ\R4'TABLE CONSISTANCY ERROR. PLEASE RE-LOAD PROGRAM'N\
     11 070264  122     065     122     X8:     .ASCIZ\R5R6'TWO UDA'S USE THE SAME VECTOR'N\
     13 070331  122     064     042     X5:     .ASCIZ\R4'CHECKSUM ERROR IN DM PROGRAM FILE 'N\
     14 070401  122     064     042     X7:     .ASCIZ\R4'ERROR IN DM PROGRAM FILE. DM PROGRAM NOT FOUND'N\
     15 070465  122     065     042     X9:     .ASCII\R5'ILLGAL CONFIGURATION FOR TEST 4'N\
     16 070531  042     103     101             .ASCIZ\"CANNONT TEST ALL UDA-50S.'N\
     45 070566  122     065     042     X13:    .ASCII\R5'MICROCODE REPORTED CONTROLLER MODEL WHICH DID NOT MATCH'N\
     46 070662  042     107     105             .ASCIZ\"GET DUST STATUS RESPONSE.  TESTING ON THIS CONTROLLER STOPS.'N\
     47 070762                          X20:
     48 070762  122     065     042     X38:    .ASCII\R5'MEMORY ERROR TRYING TO READ UDA REGISTERS'N\
     49 071040  042     103     110             .ASCII\"CHECK UNIBUS SELECTION SWITCHES ON UDA MODULE M7161 OR M7485'N\
     50 071137  042     117     122             .ASCII\"OR UNIBUS'N\
     51 071153  042     117     122             .ASCIZ\"OR "R7\
     52 071163  122     065     042     X21:    .ASCII\R5'UDA RESIDENT DIAGNOSTICS DETECTED FAILURE'NR8\
     53 071243  042     122     105             .ASCIZ\"REPLACE UDA MODULE M716'02" OR M748'03N\
     54 071314  122     065     042     X22:    .ASCII\R5'STEP BIT DID NOT SET IN UDASA REGISTER DURING INITIALIZATION'N\
     55 071415  042     123     124             .ASCIZ\"STEP BIT EXPECTED '016NR8R7\
     56 071452  122     065     042     X23A:   .ASCII\R5'UDA DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION'N\
     57 071564  104     071     042             .ASCII\D9" WORDS WERE TO BE CLEARED STARTING AT ADDRESS '016N\
     58 071652  042     106     111             .ASCII\"FIRST SEVERAL WORDS NOT CLEARED (UP TO 6):'N\
     59 071727  123     066     042             .ASCIZ\S6'ADDRESS'S4'CONTENTS'N\
     60 071760  123     067     117     X23B:   .ASCIZ\S7016S5016N\
     61 071774  122     065     042     X24:    .ASCII\R5'UDASA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION'N\
     62 072107  042     120     125             .ASCIZ\"PURGE/POLE DIAGNOSTICS WERE REQUESTED'NR8R7\
     63 072164  122     065     042     X25:    .ASCII\R5'UDA DID NOT RETURN CORRECT DATA IN UDASA REGISTER DURING INITIALIZATION'N\
     64 072300  042     040     040             .ASCIZ\"  UDASA EXPECTED  '016NR8R7\
     66 072335  122     065     042     X26:    .ASCII\R5'DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST'N\
     67 072430  042     040     040             .ASCII\"  DATA SENT TO UDASA  '016N\
     68 072464  042     040     040             .ASCIZ\"  RECEIVED FROM UDASA '016NR7\
     69 072523  122     065     042     X27:    .ASCII\R5'UDASA REGISTER DID NOT CHANGE AFTER WRITING TO IT'N\
     70 072611  042     111     116             .ASCIZ\"IN PORT LOOP DIAGNOSTIC'NR8R7\
     71 072650  122     065     042     X28:    .ASCIZ\R5'UDA DID NOT INTERRUPT THE PDP-11'NR7\
     72 072720  122     065     042     X29:    .ASCII\R5'UDA INTERRUPTED AT DIFFERENT BR LEVEL THAN SPECIFIED IN HARDWARE'N\
     73 073025  042     121     125             .ASCII\"QUESTIONS. INTERRUPT WAS AT BR LEVEL '03N\
     74 073077  042     103     110             .ASCII\"CHECK PRIORITY PLUG ON UDA MODULE M7161 OR M7485'N\
     75 073162  042     117     122             .ASCIZ\"OR CHANGE HARDWARE QUESTIONS'N\
     77 073222  122     065     042     X30:    .ASCIZ\R5'UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE RUNNING DM PROGRAM'NR8\
     78 073335  122     065     042     X31:    .ASCII\R5'NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES'N\
     79 073425  042     101     123             .ASCIZ\"ASSUME PROGRAM IS HUNG'N\
     80 073457  122     065     042     X32:    .ASCIZ\R5'MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER'N\
     81 073570  122     065     042     X35:    .ASCIZ\R5'DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE'N\
     82 073650  122     065     042     X36:    .ASCII\R5'NO INTERRUPT RECEIVED FROM UDA FOR 30 SECONDS'N\
     83 073732  042     127     110             .ASCIZ\"WHILE LOADING DM PROGRAM'N\
     84 073766  122     065     042     X37:    .ASCIZ\R5'UDA REPORTED FATAL ERROR IN UDASA REGISTER WHILE LOADING DM PROGRAM'NR8R7\
```

GLOBAL TEXT SECTION

```
      1 074103      042      115      105  XMSG1:   .ASCIZ\'MESSAGE BUFFER CONTAINS:'N\
      2 074137      123      063      117  XMSG2:   .ASCIZ\S3016S1016S1016S1016S1016S1016S1016N\
      3 074204      122      065      042  XPKT1:   .ASCII\R5'RESPONSE PACKET FROM UDA DOES NOT CONTAIN EXPECTED DATA'N\
      4 074300      042      105      111           .ASCII\'EITHER UDA RETURNED ERROR STATUS OR PACKET WAS NOT RECEIVED CORRECTLY'N\
      5 074410      123      063      042           .ASCIZ\S3'COMMAND PACKET SENT'S6'RESPONSE PACKET RECEIVED'N\
      6 074475      123      066      117  XPKT2:   .ASCIZ\S6016S1016S14016S1016N\
      7 074524      042      040      040  XSA:     .ASCIZ\'  UDASA CONTAINS  '016N\
      8 074555      042      122      105  XFRU:    .ASCIZ\'REPLACE UDA MODULE M7161 OR M7485'N\
     12                                             .EVEN
     13
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
                                ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
                                ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                                ;--
  177777                        SVCINS= -1                      ; LIST INSTRUCTIONS, SHIFTED RIGHT
  177777                        SVCTST= -1                      ; LIST TEST TAGS, SHIFTED RIGHT
  177777                        SVCSUB= -1                      ; LIST SUBTEST TAGS, SHIFTED RIGHT
  177777                        SVCGBL= -1                      ; LIST GLOBAL TAGS, SHIFTED RIGHT
  177777                        SVCTAG= -1                      ; LIST OTHER TAGS, SHIFTED RIGHT

074622                          BGNMSG ERR001
074622                                  PNTB X1,#X1A
074622  012746  067465                                                  MOV #X1A,-(SP)
074626  004137  104222                                                  JSR R1,LPNTB
074632  067564                                                          .WORD X1
074634  000002                                                          .WORD PNT.CT
074636                          ENDMSG

074640                          BGNMSG ERR002
074640                                  PNTB X2,#X2A
074640  012746  067465                                                  MOV #X2A,-(SP)
074644  004137  104222                                                  JSR R1,LPNTB
074650  067660                                                          .WORD X2
074652  000002                                                          .WORD PNT.CT
074654                          ENDMSG

074656                          BGNMSG ERR003
074656                                  PNTB X3,#X3A
074656  012746  067465                                                  MOV #X3A,-(SP)
074662  004137  104222                                                  JSR R1,LPNTB
074666  067727                                                          .WORD X3
074670  000002                                                          .WORD PNT.CT
074672                          ENDMSG

074674                          BGNMSG ERR004
074674                                  PNTB X4
074674  004137  104222                                                  JSR R1,LPNTB
074700  070012                                                          .WORD X4
074702  000000                                                          .WORD PNT.CT
074704                          ENDMSG

074706                          BGNMSG ERR005
074706                                  PNTB X5
074706  004137  104222                                                  JSR R1,LPNTB
074712  070331                                                          .WORD X5
074714  000000                                                          .WORD PNT.CT
074716                          ENDMSG

074720                          BGNMSG ERR007
074720                                  PNTB X7
074720  004137  104222                                                  JSR R1,LPNTB
074724  070401                                                          .WORD X7
074726  000000                                                          .WORD PNT.CT
074730                          ENDMSG
```

```
 38
 39 074732                          BGNMSG ERR009
 40 074732                                  PNTB X9
    074732  004137  104222                                        JSR R1,LPNTB
    074736  070465                                                .WORD X9
    074740  000000                                                .WORD PNT.CT
 41 074742                          ENDMSG
 43
 44 074744                          BGNMSG ERR006
 45 074744                                  PNTB X6
    074744  004137  104222                                        JSR R1,LPNTB
    074750  070177                                                .WORD X6
    074752  000000                                                .WORD PNT.CT
 46 074754                          ENDMSG
 47
 48 074756                          BGNMSG ERR008
 49 074756                                  PNTB X8,#X8A
    074756  012746  067465                                        MOV #X8A,-(SP)
    074762  004137  104222                                        JSR R1,LPNTB
    074766  070264                                                .WORD X8
    074770  000002                                                .WORD PNT.CT
 50 074772                          ENDMSG
 51
 57
 58 074774                          BGNMSG ERR013
 59 074774                                  PNTB X13
    074774  004137  104222                                        JSR R1,LPNTB
    075000  070566                                                .WORD X13
    075002  000000                                                .WORD PNT.CT
 60 075004                          ENDMSG
 61
 62 075006                          BGNMSG ERR020
 63 075006                                  PNTB X20
    075006  004137  104222                                        JSR R1,LPNTB
    075012  070762                                                .WORD X20
    075014  000000                                                .WORD PNT.CT
 64 075016                          ENDMSG
 65
 66 075020                          BGNMSG ERR021
 67 075020  010201                          MOV R2,R1
 68 075022  000301                          SWAB R1
 69 075024                                  AND 2,R1
    075024  042701  177775                                        BIC #^C<2>,R1
 70 075030  006201                          ASR R1
 71 075032  005201                          INC R1
 72 075034  010103                          MOV R1,R3
 73 075036  062703  000004                  ADD #4,R3
 74 075042                                  PNTB X21,R2,R1,R3
    075042  010346                                                MOV R3,-(SP)
    075044  010146                                                MOV R1,-(SP)
    075046  010246                                                MOV R2,-(SP)
    075050  004137  104222                                        JSR R1,LPNTB
    075054  071163                                                .WORD X21
    075056  000006                                                .WORD PNT.CT
 75 075060                          ENDMSG
 76
 77 075062                          BGNMSG ERR022
```

```
 78 075062  042737  100000  106762          BIC #SA.ERR,UDARSD
 79 075070                                  PNTB X22,UDARSD,R2
    075070  010246                                                  MOV R2,-(SP)
    075072  013746  106762                                          MOV UDARSD,-(SP)
    075076  004137  104222                                          JSR R1,LPNTB
    075102  071314                                                  .WORD X22
    075104  000004                                                  .WORD PNT.CT
 80 075106                                  PRINTX #XFRU
 81 075126                          ENDMSG
 82
 83 075130                          BGNMSG ERR023
 84 075130                                  PNTB X23A,R1,FFREE
    075130  013746  065016                                          MOV FFREE,-(SP)
    075134  010146                                                  MOV R1,-(SP)
    075136  004137  104222                                          JSR R1,LPNTB
    075142  071452                                                  .WORD X23A
    075144  000004                                                  .WORD PNT.CT
 85 075146  005742                          TST -(R2)
 86 075150  005712                  ERR23A: TST (R2)
 87 075152  001410                          BEQ ERR23B
 88 075154                                  PNTB X23B,R2,(R2)
    075154  011246                                                  MOV (R2),-(SP)
    075156  010246                                                  MOV R2,-(SP)
    075160  004137  104222                                          JSR R1,LPNTB
    075164  071760                                                  .WORD X23B
    075166  000004                                                  .WORD PNT.CT
 89 075170  005304                          DEC R4
 90 075172  001403                          BEQ ERR23C
 91 075174  005722                  ERR23B: TST (R2)+
 92 075176  005303                          DEC R3
 93 075200  001363                          BNE ERR23A
 94 075202                          ERR23C: PNTB XFRU
    075202  004137  104222                                          JSR R1,LPNTB
    075206  074555                                                  .WORD XFRU
    075210  000000                                                  .WORD PNT.CT
 95 075212                          ENDMSG
 96
 97 075214                          BGNMSG ERR024
 98 075214                                  PNTB X24,R2
    075214  010246                                                  MOV R2,-(SP)
    075216  004137  104222                                          JSR R1,LPNTB
    075222  071774                                                  .WORD X24
    075224  000002                                                  .WORD PNT.CT
 99 075226                          ENDMSG
100
101 075230                          BGNMSG ERR025
102 075230                                  PNTB X25,R1,R2
    075230  010246                                                  MOV R2,-(SP)
    075232  010146                                                  MOV R1,-(SP)
    075234  004137  104222                                          JSR R1,LPNTB
    075240  072164                                                  .WORD X25
    075242  000004                                                  .WORD PNT.CT
103 075244                          ENDMSG
104
106 075246                          BGNMSG ERR026
107 075246                                  PNTB X26,R2,2(R4)
    075246  016446  000002                                          MOV 2(R4),-(SP)
```

```
        075252  010246                                                  MOV R2,-(SP)
        075254  004137  104222                                          JSR R1,LPNTB
        075260  072335                                                  .WORD X26
        075262  000004                                                  .WORD PNT.CT
    108 075264                          ENDMSG
    109
    110 075266                          BGNMSG ERR027
    111 075266                                 PNTB X27,2(R4)
        075266  016446  000002                                         MOV 2(R4),-(SP)
        075272  004137  104222                                          JSR R1,LPNTB
        075276  072523                                                  .WORD X27
        075300  000002                                                  .WORD PNT.CT
    112 075302                          ENDMSG
    113
    114 075304                          BGNMSG ERR028
    115 075304                                 PNTB X28
        075304  004137  104222                                          JSR R1,LPNTB
        075310  072650                                                  .WORD X28
        075312  000000                                                  .WORD PNT.CT
    116 075314                          ENDMSG
    117
    118 075316                          BGNMSG ERR029
    119 075316                                 PNTB X29,R1
        075316  010146                                                  MOV R1,-(SP)
        075320  004137  104222                                          JSR R1,LPNTB
        075324  072720                                                  .WORD X29
        075326  000002                                                  .WORD PNT.CT
    120 075330                          ENDMSG
    122
    123 075332                          BGNMSG ERR030
    124 075332                                 PNTB X30,R1
        075332  010146                                                  MOV R1,-(SP)
        075334  004137  104222                                          JSR R1,LPNTB
        075340  073222                                                  .WORD X30
        075342  000002                                                  .WORD PNT.CT
    125 075344                          ENDMSG
    126
    127 075346                          BGNMSG ERR031
    128 075346                                 PNTB X31
        075346  004137  104222                                          JSR R1,LPNTB
        075352  073335                                                  .WORD X31
        075354  000000                                                  .WORD PNT.CT
    129 075356                          ENDMSG
    130
    131 075360                          BGNMSG ERR032
    132 075360                                 PNTB X32
        075360  004137  104222                                          JSR R1,LPNTB
        075364  073457                                                  .WORD X32
        075366  000000                                                  .WORD PNT.CT
    133 075370  004737  075562                     CALL MSGPKT
    134 075374                          ENDMSG
    135
    136 075376                          BGNMSG ERR033
    137 075376  004737  075470                     CALL PNTPKT
    138 075402                          ENDMSG
    139
    140 075404                          BGNMSG ERR034
```

```
141 075404  004737  075470                      CALL PNTPKT
142 075410                              ENDMSG
143
144 075412                              BGNMSG ERR035
145 075412                                      PNTB X35
    075412  004137  104222                                          JSR R1,LPNTB
    075416  073570                                                  .WORD X35
    075420  000000                                                  .WORD PNT.CT
146 075422  004737  075562                      CALL MSGPKT
147 075426                              ENDMSG
148
149 075430                              BGNMSG ERR036
150 075430                                      PNTB X36
    075430  004137  104222                                          JSR R1,LPNTB
    075434  073650                                                  .WORD X36
    075436  000000                                                  .WORD PNT.CT
151 075440                              ENDMSG
152
153 075442                              BGNMSG ERR037
154 075442                                      PNTB X37,R1
    075442  010146                                                  MOV R1,-(SP)
    075444  004137  104222                                          JSR R1,LPNTB
    075450  073766                                                  .WORD X37
    075452  000002                                                  .WORD PNT.CT
155 075454                              ENDMSG
156
157 075456                              BGNMSG ERR038
158 075456                                      PNTB X38
    075456  004137  104222                                          JSR R1,LPNTB
    075462  070762                                                  .WORD X38
    075464  000000                                                  .WORD PNT.CT
159 075466                              ENDMSG
160
161 075470                              PNTPKT: PNTB XPKT1
    075470  004137  104222                                          JSR R1,LPNTB
    075474  074204                                                  .WORD XPKT1
    075476  000000                                                  .WORD PNT.CT
162 075500  010401                              MOV R4,R1
163 075502  062701  000104                      ADD #HC.CPK,R1
164 075506  010402                              MOV R4,R2
165 075510  062702  000020                      ADD #HC.MPK,R2
166 075514  012703  000014                      MOV #12.,R3
167 075520                              PNTPKL: PNTB XPKT2,2(R1),(R1),2(R2),(R2)
    075520  011246                                                  MOV (R2),-(SP)
    075522  016246  000002                                          MOV 2(R2),-(SP)
    075526  011146                                                  MOV (R1),-(SP)
    075530  016146  000002                                          MOV 2(R1),-(SP)
    075534  004137  104222                                          JSR R1,LPNTB
    075540  074475                                                  .WORD XPKT2
    075542  000010                                                  .WORD PNT.CT
168 075544  062701  000004                      ADD #4,R1
169 075550  062702  000004                      ADD #4,R2
170 075554  005303                              DEC R3
171 075556  001360                              BNE PNTPKL
172 075560  000207                              RETURN
173
174 075562                              MSGPKT: PNTB XMSG1
```

```
        075562  004137  104222                                                  JSR R1,LPNTB
        075566  074103                                                          .WORD XMSG1
        075570  000000                                                          .WORD PNT.CT
    175 075572  016504  000016          MOV C.RING(R5),R4
    176 075576  062704  000272          ADD #HC.BF2,R4
    177 075602  012703  000005          MOV #5,R3
    178 075606                  MSGPKL: PNTB XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)
        075606  016446  000014                                                  MOV 12.(R4),-(SP)
        075612  016446  000012                                                  MOV 10.(R4),-(SP)
        075616  016446  000010                                                  MOV 8.(R4),-(SP)
        075622  016446  000006                                                  MOV 6(R4),-(SP)
        075626  016446  000004                                                  MOV 4(R4),-(SP)
        075632  016446  000002                                                  MOV 2(R4),-(SP)
        075636  011446                                                          MOV (R4),-(SP)
        075640  004137  104222                                                  JSR R1,LPNTB
        075644  074137                                                          .WORD XMSG2
        075646  000016                                                          .WORD PNT.CT
    179 075650  062704  000016          ADD #14.,R4
    180 075654  005303                  DEC R3
    181 075656  001353                  BNE MSGPKL
    182 075660  000207                  RETURN
```

```
075662                          BGNMSG ERRRTN                          ;ERROR REPORT ROUTINE
075662  013702  065050                 MOV TNUM,R2                     ;GET TEST NUMBER
075666  006302                         ASL R2                          ;DOUBLE
075670  012703  067436                 MOV #BASL3,R3                   ;GET ADDRESS OF DRIVE PRINT LINE
075674  005764  000004                 TST 4(R4)                       ;CHECK IF DRIVE NUMBER GIVEN
075700  100002                         BPL 1$                          ;BRANCH IF SO
075702  012703  067453                 MOV #BAS,R3
075706                          1$:    PNTB BASLN,TNAMES-2(R2),#BASL1,(R4),#BASL2,(R5),R3,4(R4)
075706  016446  000004                                                         MOV 4(R4),-(SP)
075712  010346                                                                 MOV R3,-(SP)
075714  011546                                                                 MOV (R5),-(SP)
075716  012746  067417                                                         MOV #BASL2,-(SP)
075722  011446                                                                 MOV (R4),-(SP)
075724  012746  067401                                                         MOV #BASL1,-(SP)
075730  016246  102520                                                         MOV TNAMES-2(R2),-(SP)
075734  004137  104222                                                         JSR R1,LPNTB
075740  067454                                                                 .WORD BASLN
075742  000016                                                                 .WORD PNT.CT
075744                                         ASSUME C.UADR EQ 0
075744  004737  107614                 CALL RNTIME                     ;GET RUNTIME PARAMETERS
075750                                 PRINT #CR                       ;ADVANCE TO NEW LINE
075750  112700  000015                                                         MOVB #CR,R0
075754  004737  104040                                                         CALL CPNT
075760  062704  000006                 ADD #6,R4                       ;INCREASE R4 TO POINT TO MESSAGE POINTER
075764  012402                         MOV (R4)+,R2                    ;GET MESSAGE POINTER
075766  006302                         ASL R2                          ;DOUBLE TO MAKE BYTE OFFSET
075770  063702  065034                 ADD DMPROG,R2                   ;ADD TO START OF MESSAGE STRINGS
075774  067702  167034                 ADD @DMPROG,R2                  ;ADD SIZE OF MAIN PROGRAM
076000  105712                         TSTB (R2)                       ;CHECK FIRST BYTE
076002  001001                         BNE NCON                        ;IF ZERO
076004  005202                         INC R2                          ; INCREMENT TO NEXT BYTE
076006  012737  104140  065012  NCON:  MOV #PX,PTYPE                   ;CHANGE TO EXTENDED OUTPUT
076014  004737  102220                 CALL OSTRNG                     ;OUTPUT ACCORDING TO STRING
076020                          ENDMSG
```

GLOBAL ERROR REPORT SECTION

```
1       000001          SVCINS= 1               ; LIST INSTRUCTIONS, SHIFTED RIGHT
2       000001          SVCTST= 1               ; LIST TEST TAGS, SHIFTED RIGHT
3       000001          SVCSUB= 1               ; LIST SUBTEST TAGS, SHIFTED RIGHT
4       000001          SVCGBL= 1               ; LIST GLOBAL TAGS, SHIFTED RIGHT
5       000001          SVCTAG= 1               ; LIST OTHER TAGS, SHIFTED RIGHT
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;MEMORY ALLOCATION ERROR
                                ;
                                ;THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST

076022                          FMERR:  ERRSF 4,,ERR004
076022  104454                                                          TRAP    C$ERSF
076024  000004                                                          .WORD   4
076026  000000                                                          .WORD   0
076030  074674                                                          .WORD   ERR004
076032                                  DOCLN                   ;ABORT
076032  104444                                                          TRAP    C$DCLN
```

```
      1                                 ;ALOCM
      2                                 ;
      3                                 ;ALLOCATE A BLOCK OF FREE MEMORY.  REPORT ERROR IF MEMORY EXHAUSTED.
      4                                 ;
      5                                 ;INPUTS:
      6                                 ;       R1 - NUMBER OF WORDS TO ALLOCATE
      7                                 ;       FFREE - FIRST FREE WORD IN MEMORY
      8                                 ;       FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
      9                                 ;OUTPUTS:
     10                                 ;       R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
     11                                 ;       FFREE - NEW FIRST FREE WORD IN MEMORY
     12                                 ;       FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
     13                                 ;SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
     14                                 ;AND ENTIRE PROGRAM WILL BE STOPPED.
     15
     16 076034                          ALOCM:  PUSH FFREE                      ;SAVE FFREE AT ENTRY
        076034  013746  065016                                                          MOV FFREE,-(SP)
     17 076040  160137  065020                  SUB R1,FSIZE                    ;REDUCE SIZE OF FREE MEMORY
     18 076044  002766                          BLT FMERR                       ;REPORT ERROR IF NOT ENOUGH MEMORY
     19 076046  060101                          ADD R1,R1                       ;CHANGE WORDS TO BYTES
     20 076050  060137  065016                  ADD R1,FFREE                    ;CALCULATE NEW START OF FREE MEMORY
     21 076054                                  POP R1                          ;GET START OF ALLOCATED MEMORY
        076054  012601                                                                  MOV (SP)+,R1
     22 076056  000207                          RETURN
```

```
                                        ;HCOMM
                                        ;
                                        ;ALLOCATES MEMORY FOR HOST COMM AREA AND PACKET BUFFERS WITH ONE
                                        ;DESCRIPTOR IN EACH RING. TO BE CALLED AFTER INITIALIZING
                                        ;A CONTROLLER WITH SA.MSG=0 AND SA.CMD=0.
                                        ;
                                        ;INPUTS:
                                        ;       R5 - ADDRESS OF CONTROLLER TABLE
                                        ;OUTPUTS:
                                        ;       CONTROLLER TABLE POINTING TO HOST COMM AREA
                                        ;       RING POINTERS TO PACKETS
                                        ;       R4 - ADDRESS OF HOST COMM AREA

076060  012701  000200        HCOMM:  MOV #HC.SIZ/2,R1                ;GET SIZE OF AREA TO ALLOCATE
076064  004737  076034                CALL ALOCM                      ;ALLOCATE THE MEMORY
076070  010104                        MOV R1,R4                       ;GET ADDRESS OF HOST COMM AREA
076072  010465  000016                MOV R4,C.RING(R5)               ;PLACE IN CONTROLLER TABLE
076076  062701  000020                ADD #HC.MPK,R1                  ;COMPUTE START OF MESSAGE PACKET
076102  010164  000004                MOV R1,HC.MSG(R4)               ;PLACE IN RING
076106  062701  000064                ADD #<HC.CPK-HC.MPK>,R1         ;COMPUTE START OF COMMAND PACKET
076112  010164  000010                MOV R1,HC.CMD(R4)               ;PLACE IN RING
076116  000207                        RETURN
```

```
                                        ;TINIT
                                        ;
                                        ;INITIALIZE VARIABLES FOR TEST
                                        ;
                                        ;INPUTS:
                                        ;       R1 - TEST NUMBER
                                        ;OUTPUTS:
                                        ;       LBUFS - CLEARED (DELETES ERROR LOG)
                                        ;       TNUM - TEST NUMBER FROM R1
                                        ;       FNUM - LAST LOADED TEST IN TNUM < 4
                                        ;       ALL REGISTERS CLOBERED
                                        ;
076120  010137  065050          TINIT:  MOV R1,TNUM                     ;SAVE TEST NUMBER
076124  004737  107474                  CALL RESET                      ;RESET ALL DEVICES
076130  005037  065254                  CLR LBUFS                       ;CLEAR ERROR LOG BUFFER POINTER
076134  013737  065022  065016          MOV FMEM,FFREE                  ;INIT FREE
076142  013737  065024  065020          MOV FMEMS,FSIZE                 ;INIT FSIZE
076150  022701  000004                  CMP     #4,R1                   ;ARE WE DOING TEST 4?
076154  001431                          BEQ     TIEXIT                  ; IF SO, EXIT
076156  006301                          ASL     R1                      ; ELSE, CHECK IF FILE IS LOADED
076160  020137  065046                  CMP     R1,FNUM                 ; IF FILE ALREADY IN MEMORY?
076164  001425                          BEQ     TIEXIT                  ; IF SO, EXIT
076166  005301                          DEC     R1                      ; ELSE, GET FILE NUMBER
076170  012705  001122                  MOV     #<STORAG-DMFRST>,R5     ; R5->ADDRESS TO STORE - DM FIRST ADDRESS
076174  012737  002122  065034          MOV     #STORAG,DMPROG          ; SAVE DMPROG ADDRESS
076202  004737  107044                  CALL    RDREC                   ; READ IN RECORD
076206  103415                          BCS     TINITE                  ; IF ERROR, REPORT
076210  063705  002122                  ADD     STORAG,R5               ;
076214  063705  002126                  ADD     STORAG+4,R5             ; R5 -> PAST 1ST DLL OVERLAY (MESSAGES)
076220  005201                          INC     R1                      ; R1 = NEXT FILE TO READ
076222  004737  107044                  CALL    RDREC                   ; READ FILE
076226  103405                          BCS     TINITE                  ; IF ERROR, REPORT
076230  062705  001000                  ADD     #DMFRST,R5              ; R5 -> UDA50 DM PROGRAM
076234  010537  065036                  MOV     R5,D50PRG               ; AND SAVE
076240  000207                  TIEXIT: RETURN
                                        
076242                          TINITE: ERRSF 7,,ERR007                 ;REPORT DM PROGRAM NOT FOUND
076242  104454                                                                  TRAP    C$ERSF
076244  000007                                                                  .WORD   7
076246  000000                                                                  .WORD   0
076250  074720                                                                  .WORD   ERR007
076252                                  DOCLN
076252  104444                                                                  TRAP    C$DCLN
```

```
                                        ;RNT4DM
                                        ;
                                        ;LOAD AND RUN A TEST 4 IN THE CONTROLLERS. RETURN WHEN ALL
                                        ;DM PROGRAMS HAVE TERMINATED.  ERROR WILL OCCUR IF DRIVE
                                        ;ATTACHED TO UDA52 IS THE LOAD DEVICE.
                                        ;
                                        ;INPUTS:
                                        ;       TSTTAB - POINTER TO FIRST CONTROLLER TABLE
                                        ;       R1 - NUMBER OF CONTROLLERS TO TEST
                                        ;OUTPUTS:
                                        ;       DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
                                        ;       Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
                                        ;ALL REGISTERS ARE USED AND PREVIOUS CONTENTS DESTROYED.
 076254                                 RNT4DM:
 076254  005037  065054                         CLR     URNING
 076260  005037  065136                         CLR     TYPCNT                          ; CLEAR FLAGS
 076264  010137  065052                         MOV     R1,URUN                         ; URUN = # OF UNITS
 076270  012737  002122  065034                 MOV     #STORAG,DMPROG                  ; DMPROG -> WHERE EITHER TEST 4 IS LOADED
 076276  013737  065052  065056                 MOV     URUN,UCNT
 076304  013705  065032                         MOV     TSTTAB,R5                       ; R5 -> CONTROLLER TABLE
                                        ; GET DUST STATUS FOR ALL CONTROLLERS & STORE IN CONTROLLER TABLES
 076310  005765  000002                 TSTCNT: TST     C.UNIT(R5)
 076314  100435                                 BMI     RNT42                           ; IF NOT TO TEST, BRANCH
 076316  004737  105202                         CALL    GTDUST                          ; GET DUST STATUS
 076322  023764  065132  000034                 CMP     U52EXT,HC.MPK+P.DEXT(R4) ; IS IT A 52?
 076330  001020                                 BNE     RNT41                           ; IF NOT,BRANCH
 076332  052737  000001  065136                 BIS     #TY.U52,TYPCNT                  ; IF SO, SET 52 CONTROLLER
 076340  032765  000040  000014                 BIT     #CT.U50,C.FLG(R5)
 076346  001420                                 BEQ     RNT42
 076350                                 RNT4E:  ERRDF   13,,ERR013
 076350  104455                                                                         TRAP    C$ERDF
 076352  000015                                                                        .WORD   13
 076354  000000                                                                        .WORD   0
 076356  074774                                                                        .WORD   ERR013
 076360                                         DORPT
 076360  104424                                                                        TRAP    C$DRPT
 076362  052765  100000  000002                 BIS     #BIT15,C.UNIT(R5)       ; STOP TESTING ON THAT UNIT
AU 076370  000777                               BR      RNTT42                  ; END CONTINUE
 076372  052737  000002  065136  RNT41:  BIS     #TY.U50,TYPCNT                  ; SET 50 CONTROLLER
 076400  032765  000040  000014                 BIT     #CT.U50,C.FLG(R5)
 076406  001760                                 BEQ     RNT4E
 076410  062705  000046          RNT42:  ADD     #C.SIZE,R5
 076414  005337  065056                         DEC     UCNT
 076420  001333                                 BNE     TSTCNT                  ; CONTINUE UNTIL DONE

                                        ; NOW ALL CONTROLLER TYPES ARE KNOWN, CHECK IF THERE IS MIX
 076422  022737  000003  065136                 CMP     #<TY.U50+TY.U52>,TYPCNT ; WERE THERE TWO TYPES OF CONTROLLERS?
 076430  001413                                 BEQ     RNT4DT                  ; IF SO, GO GET BOTH CODES
                                        ; IF HERE, ONLY 1 TYPE OF CONTROLLER
 076432  032737  000001  065136                 BIT     #TY.U52,TYPCNT          ; ARE THEY ALL 52'S ?
 076440  001003                                 BNE     3$                      ; IF SO, BRANCH
                                        ; IF HERE, ALL ARE 50'S
 076442  004737  076770                         CALL    GTT450                  ; GET TEST 4 FOR U50
 076446  000402                                 BR      4$                      ;
 076450  004737  076762          3$:     CALL    GTT452                  ; GET TEST 4 FOR U52
 076454  000137  077032          4$:     JMP     STLDDM                  ; GO START LOADING DM PROGRAMS
```

```
                                        ;THERE IS A MIX OF CONTROLLERS, NOW DO THE FOLLOWING
                                        ;       1) INIT 52 CONTROLLERS
                                        ;       2) READ 52 DM PROGRAM
                                        ;       3) IF ANY 52 NOT AT INIT STATE -- ERROR
                                        ;       4) LOAD 52 CONTROLLERS
                                        ;       5) READ 50 DM PROG
                                        ;       6) LOAD 50 CONTROLLERS
                                        ;
076460  013737  065052  065056  RNT4DT: MOV     URUN,UCNT
076466  013705  065032                  MOV     TSTTAB,R5
076472  005000                          CLR     R0                      ; INIT NUMBER OF 52'S
                                        ; INIT 52'S
076474  005765  000002          1$:     TST     C.UNIT(R5)
076500  100406                          BMI     2$                      ; DON'T TEST CONTROLLER IF DROPPED
076502  032765  000040  000014          BIT     #CT.U50,C.FLG(R5)       ; IS IT A 50?
076510  001002                          BNE     2$                      ; IF SO, BRANCH
076512  005075  000000                  CLR     @C.UADR(R5)             ; 52 IF HERE, INIT 52
076516  062705  000046          2$:     ADD     #C.SIZE,R5              ; GO TO NEXT CONTROLLER
076522  005337  065056                  DEC     UCNT
076526  001362                          BNE     1$
                                        ; GET 52 CODE
076530  004737  076762                  CALL    GTT452
                                        ; IF ANY CONTROLLER NOT IN INIT STATE -> ERROR
076534  013737  065052  065056          MOV     URUN,UCNT               ; # OF CONTROLLERS
076542  013705  065032                  MOV     TSTTAB,R5               ; START OF CONTROLLER TABLES
076546  016504  000000          6$:     MOV     C.UADR(R5),R4           ; R4 -> UDAIP
076552  042765  177737  000014          BIC     #^C<CT.U50>,C.FLG(R5)   ; CLEAR FLAGS
076560  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN
076566  005765  000002                  TST     C.UNIT(R5)
076572  100416                          BMI     8$                      ; IF CONTROLLER DROPPED, DON'T TEST
076574  032765  000040  000014          BIT     #CT.U50,C.FLG(R5)       ; IS IT A 50?
076602  001012                          BNE     8$                      ; IF SO, BRANCH
076604  005764  000002                  TST     2(R4)                   ; IS IT INITED?
076610  001002                          BNE     7$                      ; IF SO, CONTINUE
076612  000137  076744                  JMP     RNT4DE                  ; ELSE, ERROR
                                        ; LOAD 52 CODE
076616  004737  104456          7$:     CALL    LOADDM                  ; LOAD 52 CODE
076622  001402                          BEQ     8$
076624  005237  065054                  INC     URNING
076630  062705  000046          8$:     ADD     #C.SIZE,R5
076634  005337  065056                  DEC     UCNT
076640  001342                          BNE     6$                      ; CONTINUE UNTIL DONE
                                        ; GET 50 CODE
076642  004737  076770          RNT450: CALL    GTT450
                                        ; LOAD 50 CODE
076646  013737  065052  065056          MOV     URUN,UCNT
076654  013705  065032                  MOV     TSTTAB,R5
076660  005765  000002          10$:    TST     C.UNIT(R5)
076664  100417                          BMI     11$
076666  032765  000040  000014          BIT     #CT.U50,C.FLG(R5)       ; IS IT A 50?
076674  001413                          BEQ     11$                     ; IF NOT, BRANCH
076676  042765  177737  000014          BIC     #^C<CT.U50>,C.FLG(R5)   ; ELSE, CLEAR BITS
076704  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; SAVE LOGIC UNIT NUMBER
076712  004737  104456                  CALL    LOADDM                  ; LOAD DM CODE
076716  001402                          BEQ     11$                     ; IF ERROR, BRANCH
076720  005237  065054                  INC     URNING                  ; ELSE, CONTROLLER IS RUNNING
076724  062705  000046          11$:    ADD     #C.SIZE,R5              ; POINT TO NEXT CONTROLLER TABLE
```

```
     58 076730  005337  065056          DEC     UCNT
     59 076734  001351                  BNE     10$                     ; CONTINUE UNTIL DONE
     60 076736  005737  065054          TST     URNING                  ; SET FLAG BY CHECKING IF ANY CONTROLLERS RUN
     61 076742  000207                  RETURN
```

GLOBAL SUBROUTINES SECTION

```
      1 076744                          RNT4DE: ERRSF   9,,ERR009
        076744  104454                                                  TRAP    C$ERSF
        076746  000011                                                  .WORD   9
        076750  000000                                                  .WORD   0
        076752  074732                                                  .WORD   ERR009
      2 076754  000261                    SEC
      3 076756  000137  076642            JMP     RNT450          ; START LOADING 50 CODE
```

```
                                        ;GTT452 & GTT450
                                        ;
                                        ;GET TEST 4 FOR UDA50 OR UDA52 (DEPENDING ON THE ENTRY POINT)
                                        ;
076762  012701  000010        GTT452:   MOV     #8.,R1                  ; R1 = T4 FOR 52 FNUM
076766  000402                          BR      GTT4
076770  012701  000011        GTT450:   MOV     #9.,R1                  ; R1 = T4 FOR 50 FNUM
076774  012705  001122        GTT4:     MOV     #<STORAG-DMFRST>,R5
077000  020137  065046                  CMP     R1,FNUM                 ; DMPROG ALREADY IN MEMORY?
077004  001405                          BEQ     1$                      ; IF SO, EXIT
077006  004737  107044                  CALL    RDREC                   ; ELSE, READ RECORD.
077012  103002                          BCC     1$
077014  000137  076242                  JMP     TINITE                  ; BRANCH IF ERROR
077020  000207                1$:       RETURN
```

```
                                ;RUNDM
                                ;
                                ;LOAD AND RUN A DM PROGRAM IN THE CONTROLLERS. RETURN WHEN ALL
                                ;DM PROGRAMS HAVE TERMINATED.
                                ;
                                ;INPUTS:
                                ;       TSTTAB - POINTER TO FIRST CONTROLLER TABLE
                                ;       R1 - NUMBER OF CONTROLLERS TO TEST
                                ;IMPLICIT INPUTS:
                                ;       DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
                                ;OUTPUTS:
                                ;       Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
                                ;ALL REGISTERS ARE USED AND PREVIOUS CONTENTS DESTROYED.

 077022  010137  065052         RUNDM:  MOV R1,URUN                     ;SAVE NUMBER OF UNITS TO RUN
 077026  005037  065054                 CLR URNING                      ;CLEAR NUMBER OF UNITS RUNNING

                                ;LOAD DM PROGRAM INTO EACH CONTROLLER

 077032  013737  065052  065056 STLDDM: MOV URUN,UCNT                   ;SET COUNTER OF UNITS
 077040  013705  065032                 MOV TSTTAB,R5                   ;GET FIRST CONTROLLER TABLE
 077044                         LDDM:
 077044  042765  177737  000014         BIC #^C<CT.U50>,C.FLG(R5)               ;CLEAR ALL FLAGS
 077052  116537  000002  002074         MOVB C.UNIT(R5),L$LUN           ;SEE IF UNIT TO BE TESTED
 077060  005765  000002                 TST C.UNIT(R5)
 077064  100405                        BMI LDNEXT                      ;IF NOT, DON'T LOAD THIS UNIT
 077066                                         ASSUME CT.AVL EQ BIT15
 077066  004737  104456                 CALL LOADDM                     ;LOAD THE DM PROGRAM
 077072  001402                        BEQ LDNEXT                      ;IF ERROR, GO TO NEXT CONTROLLER
 077074  005237  065054                 INC URNING                      ;IF NO ERROR, COUNT UNIT RUNNING
 077100  062705  000046         LDNEXT: ADD #C.SIZE,R5                  ;MOVE TO NEXT CONTROLLER TABLE
 077104  005337  065056                 DEC UCNT                        ;CHECK IF MORE CONTROLLERS
 077110  001355                        BNE LDDM                        ;LOAD NEXT

                                ;CHECK IF ANY CONTROLLERS LOADED

 077112  005737  065054                 TST URNING                      ;ANY UNITS LOADED?

                                ;THE DM PROGRAMS ARE NOW IN CONTROL
                                ;RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS

 077116  000207                        RETURN
```

```
                                        ;RESPDM
                                        ;
                                        ;RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
                                        ;HAVE TERMINATED.
                                        ;
 077120  013705  065032                 RESPDM: MOV TSTTAB,R5                   ;GET CONTROLLER TABLE ADDRESS
 077124  013737  065052  065056                 MOV URUN,UCNT                   ;SET COUNTER OF UNITS
 077132  016504  000016                 RESPCT: MOV C.RING(R5),R4               ;GET HOST COMM AREA ADDRESS
 077136  032765  000002  000014                 BIT #CT.RN,C.FLG(R5)            ;CHECK IF PROGRAM RUNNING
 077144  001446                                 BEQ RSPNXT                      ;IF NOT, LOOK AT NEXT
 077146  116537  000002  002074                 MOVB C.UNIT(R5),L$LUN           ;STORE UNIT NUMBER UNDER TEST
 077154  032765  000010  000014                 BIT #CT.MSG,C.FLG(R5)           ;SEE IF INTERRUPT RECEIVED
 077162  001071                                 BNE RSPIN                       ;IF SO, LOOK AT PACKET
 077164  032765  000004  000014                 BIT #CT.CMD,C.FLG(R5)           ;SEE IF COMMAND HAS BEEN SENT
 077172  001520                                 BEQ RSPOU                       ;IF NOT, SEND ONE

                                        ;CHECK IF UDA STILL RUNNING

 077174  011503                                 MOV (R5),R3                     ;GET ADDRESS OF UDAIP
 077176  016301  000002                         MOV 2(R3),R1                    ;LOOK AT UDASA REGISTER
 077202  001405                                 BEQ RSPTM                       ;IF ZERO, UDA STILL RUNNING
 077204                                         ERRDF 30,,ERR030                ;REPORT UDA HAS FATAL ERROR
 077204  104455                                                                         TRAP    C$ERDF
 077206  000036                                                                        .WORD   30
 077210  000000                                                                        .WORD   0
 077212  075332                                                                        .WORD   ERR030
 077214  000445                                 BR RSPDRP                       ;DROP CONTROLLER FROM TESTING

                                        ;CHECK FOR TIMEOUT OF RESPONSE

 077216                                 RSPTM:
 077216  005737  065222                         TST KW.CSR                      ;SEE IF A CLOCK ON SYSTEM
 077222  001416                                 BEQ RSPNTO                      ;DON'T TIME IF NO CLOCK
 077224  023765  065234  000042                 CMP KW.EL+2,C.TOH(R5)           ;COMPARE TO TIMEOUT COUNTER
 077232  101005                                 BHI RSPTMO
 077234  001011                                 BNE RSPNTO
 077236  023765  065232  000040                 CMP KW.EL,C.TO(R5)
 077244  103405                                 BLO RSPNTO
 077246                                 RSPTMO: ERRDF 31,,ERR031                ;IF TOO MUCH TIME ELAPSED SINCE LAST INTERRUPT
 077246  104455                                                                 ;REPORT TIMEOUT ERROR
 077250  000037                                                                        TRAP    C$ERDF
 077252  000000                                                                        .WORD   31
 077254  075346                                                                        .WORD   0
 077256  000424                                                                        .WORD   ERR031
                                         BR RSPDRP                       ;DROP CONTROLLER FROM TESTING
 077260                                 RSPNTO:
 077260                                         BREAK                           ;ALLOW DRS TO SEE TERMINAL INPUT
 077260  104422                                                                         TRAP    C$BRK
```

```
                                        ;CHECK FOR TIME TO PRINT STATISTICAL REPORT
                                        
077262  005737  065222          RSPNXT: TST KW.CSR                      ;ANY CLOCK ON SYSTEM?
077266  001412                          BEQ RSPNRP                      ;BYPASS IF NOT
077270  023737  065234  065240          CMP KW.EL+2,STIME+2             ;  A STATISTICAL REPORT
077276  101005                          BHI RSPRPT
077300  001005                          BNE RSPNRP
077302  023737  065232  065236          CMP KW.EL,STIME
077310  103401                          BLO RSPNRP
077312                          RSPRPT: DORPT                           ;PRINT THE REPORT
077312  104424                                                                  TRAP    C$DRPT
                                        
                                        ;SWITCH TO NEXT CONTROLLER
                                        
077314  062705  000046          RSPNRP: ADD #C.SIZE,R5                  ;MOVE TO NEXT TABLE
077320  005337  065056                  DEC UCNT                        ;CHECK IF MORE CONTROLLERS
077324  001302                          BNE RESPCT                      ;LOOK AT NEXT CONTROLLER
077326  000674                          BR RESPDM                       ;LOOK AT FIRST CONTROLLER AGAIN
                                        
                                        ;REMOVE A CONTROLLER FROM TESTING
                                        
077330  042765  000012  000014  RSPDRP: BIC #CT.RN+CT.MSG,C.FLG(R5)      ;CLEAR PROGRAM RUNNING
077336  005337  065054                  DEC URNING                      ;REDUCE RUNNING CONTROLLERS COUNT
077342  001347                          BNE RSPNXT                      ;IF ANY STILL RUNNING, LOOK AT THEM
077344  000207                          RETURN                          ;ELSE RETURN TO TEST SECTION
```

```
                                        ;CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
                                        ;
                                        ;CHECK FOR PROPER OPCODE IN END PACKET
                                        ;
077346  012700  000204          RSPIN:  MOV #OP.END+OP.SSD,R0           ;GET SEND DATA END PACKET OPCODE
077352  032765  000020  000014          BIT #CT.REQ,C.FLG(R5)           ;LOOK IF SEND DATA OR RECEIVE DATA
077360  001402                          BEQ RSPMWR
077362  012700  000205                  MOV #OP.END+OP.RSD,R0           ;CHANGE TO RECEIVE DATA END PACKET OPCODE
077366  120064  000030          RSPMWR: CMPB R0,HC.MPK+P.OPCD(R4)       ;COMPARE TO OPCODE IN END PACKET
077372  001010                          BNE RSPERR
                                        ;
                                        ;LOOK AT STATUS CODE
                                        ;
077374  032764  000037  000032          BIT #ST.MSK,HC.MPK+P.STS(R4)    ;CHECK FOR STATUS CODE ST.SUC (ZERO)
077402  001004                          BNE RSPERR
                                        ;
                                        ;CHECK FOR EXPECTED REFERENCE NUMBER
                                        ;
077404  026564  000044  000020          CMP C.REF(R5),HC.MPK+P.CRF(R4)  ;CHECK IF CORRECT REF NUMBER
077412  001405                          BEQ RSPPTW
077414                          RSPERR: ERRDF 33,,ERR033
077414  104455                                                          TRAP    C$ERDF
077416  000041                                                          .WORD   33
077420  000000                                                          .WORD   0
077422  075376                                                          .WORD   ERR033
077424  000741                          BR RSPDRP                       ;DROP UNIT FROM TESTING
                                        ;
                                        ;CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
                                        ;
077426  032765  000020  000014  RSPPTW: BIT #CT.REQ,C.FLG(R5)           ;CHECK IF RESPONSE FROM DM PROGRAM
077434  001445                  RSPOU:  BEQ RSPOUT                      ;LOOK AT REQUEST NUMBER IF SO
```

GLOBAL SUBROUTINES SECTION

```
                                        ;MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
                                        
077436  016401  000272          RSPPT2: MOV HC.BF2(R4),R1               ;GET REQUEST NUMBER
077442  042701  007777                  BIC     #007777,R1              ;CHECK TYPE
077446  022701  060000                  CMP     #DU.SPC,R1              ;IS SPECIAL TYPE SET?
077452  001010                          BNE     1$                      ;IF NOT, ERROR
077454  042764  170000  000272          BIC     #^C007777,HC.BF2(R4)    ;CLEAR TYPE
077462  016401  000272                  MOV HC.BF2(R4),R1               ;GET REQUEST NUMBER
077466  020127  000017                  CMP R1,#DSPSIZ                  ;CHECK IF IN EXPECTED RANGE
077472  103405                          BLO RSPPT3
077474                          1$:     ERRDF 32,,ERR032                ;BAD REQUEST NUMBER
077474  104455                                                                  TRAP    C$ERDF
077476  000040                                                                  .WORD   32
077500  000000                                                                  .WORD   0
077502  075360                                                                  .WORD   ERR032
077504  000711                          BR RSPDRP                       ;DROP UNIT FROM TESTING
                                        
077506  012700  000004          RSPPT3: MOV #OP.SSD,R0                  ;BUILD A SEND DATA COMMAND PACKET
077512  004737  105302                  CALL BLDCMD                     ;  FOR ANSWER TO DM PROGRAM
077516  012700  000164                  MOV #HC.BF1,R0                  ;POINT TO BUFFER IN PACKET
077522  004737  105464                  CALL CLRBUF                     ; AND CLEAR BUFFER
077526  010403                          MOV R4,R3                       ;R3 POINTS TO COMMAND BUFFER
077530  062704  000106                  ADD #HC.BSZ,R4                  ;R4 POINTS TO MESSAGE BUFFER
077534  011401                          MOV (R4),R1                     ;GET REQUEST NUMBER
077536  012423                          MOV (R4)+,(R3)+                 ;PUT REQUEST NUMBER INTO COMMAND PACKET
077540  060101                          ADD R1,R1                       ;DOUBLE REQUEST NUMBER
077542  004771  077652                  CALL @RSPDSP(R1)                ;CALL REQUESTED ROUTINE
077546  001270                          BNE RSPDRP                      ;ROUTINE RETURNS Z CLEAR TO DROP UNIT FROM TESTING
                                                                        ; Z SET IF COMMAND READY TO SEND TO UNIT
                                        
                                        ;SEND COMMAND BACK TO UDA
                                        
077550  042765  000010  000014 RSPOUT: BIC #CT.MSG,C.FLG(R5)            ;CLEAR MESSAGE RECEIVED FLAG
077556  032765  000020  000014          BIT #CT.REQ,C.FLG(R5)           ;CHECK WHICH COMMAND TO SEND
077564  001014                          BNE RSPOU2                      ;BRANCH IF RESPONSE TO REQUEST
                                        
077566  012700  000005                  MOV #OP.RSD,R0                  ;BUILD RECEIVE DATA COMMAND
077572  004737  105302                  CALL BLDCMD
077576  012700  000272                  MOV #HC.BF2,R0                  ;POINT TO MESSAGE BUFFER
077602  004737  105464                  CALL CLRBUF                     ;  AND CLEAR IT
077606  052765  000020  000014          BIS #CT.REQ,C.FLG(R5)           ;SET REQUEST BIT
077614  000403                          BR RSPOU3
                                        
077616  042765  000020  000014 RSPOU2: BIC #CT.REQ,C.FLG(R5)           ;CLEAR REQUEST BIT
077624                          RSPOU3:
077624  004737  105366                  CALL SNDCMD                     ;SEND COMMAND TO UDA
077630  012700  000264                  MOV #3.*60.,R0                  ;SET TIMEOUT FOR 3 MINUTES
077634  010501                          MOV R5,R1
077636  062701  000040                  ADD #C.TO,R1                    ;PUT TIME IN CONTROLLER TABLE
077642  004737  105736                  CALL SETTO
077646  000137  077262                  JMP RSPNXT                      ;NOW WAIT FOR END PACKET
```

```
                                ;RESPONSE REQUEST DISPATCH TABLE

077652  077710          RSPDSP: .WORD T1MSIZ            ; 0. SET UP FREE MEMORY FOR ADDRESS TESTING
077654  100030                  .WORD T2DLL             ; 1. PROVIDE DIAGNOSTIC PROGRAM FOR DISK DRIVE
077656  100174                  .WORD T2CMD             ; 2. GET MANUAL INTERVENTION COMMAND
077660  100644                  .WORD T4MPRM            ; 3. TELL DATA PATTERN 16
077662  100666                  .WORD T4UPRM            ; 4. TELL UNIT PARAMETERS, CLEAR CONTENTS
077664  101146                  .WORD T4BB1             ; 5. TELL BAD BLOCKS (FIRST 14)
077666  101176                  .WORD T4BB2             ; 6. TELL BAD BLOCKS (LAST TWO)
077670  101226                  .WORD T4SOFT            ; 7. ADD TO SOFT ERROR AND ECC COUNTS
077672  101254                  .WORD T4SEEK            ; 8. ADD 1000 TO SEEK COUNT
077674  101274                  .WORD T4MXFR            ; 9. ADD TO MEGABITS READ AND WRITE COUNTS
077676  101436                  .WORD UTOTST            ;10. TELL WHICH DRIVES TO TEST
077700  101542                  .WORD ERRMES            ;11. REPORT ERROR MESSAGE
077702  101762                  .WORD ERRMC             ;12. REPORT ERROR MESSAGE AND COUNT HARD ERROR
077704  102102                  .WORD MESSAG            ;13. PRINT A DESCRIPTIVE MESSAGE
077706  102214                  .WORD DONE              ;14. MARK DM PROGRAM AS NO LONGER RUNNING

        000017                  DSPSIZ=<.-RSPDSP>/2     ;LEGAL NUMBERS ARE LOWER THAN THIS
```

```
                        ;NORMAL MAINTENANCE READ BUFFER DESCRIPTION

                        ;BYTE OFFSET FROM
                        ;START OF BUFFER
                        ;
                        ;       0       +-------------------------------+
                        ;               !       REQUEST NUMBER          !  USED TO SELECT ROUTINE
                        ;       2       +-------------------------------+
                        ;               !       DATA ARGUMENT #1        !  R4 CONTAINS THIS ADDRESS
                        ;       4       +-------------------------------+
                        ;               !       DATA ARGUMENT #2        !
                        ;       6       +-------------------------------+
                        ;               !       DATA ARGUMENT #3        !
                        ;       8       +-------------------------------+
                        ;               !       DATA ARGUMENT #4        !
                        ;       10      +-------------------------------+
                        ;               !       DATA ARGUMENT #5        !
                        ;       12      +-------------------------------+
                        ;               !       DATA ARGUMENT #6        !
                        ;       14      +-------------------------------+
                        ;               !       DATA ARGUMENT #7        !
                        ;       16      +-------------------------------+
                        ;               !       DATA ARGUMENT #8        !
                        ;       18      +-------------------------------+
                        ;               !       DATA ARGUMENT #9        !
                        ;       20      +-------------------------------+
                        ;               !       DATA ARGUMENT #10       !
                        ;       22      +-------------------------------+
                        ;               !       DATA ARGUMENT #11       !
                        ;               +-------------------------------+
                        ;
                        ;       .               .
                        ;       .               .
                        ;       .               .
                        ;
                        ;       68      +-------------------------------+
                        ;               !       DATA ARGUMENT #34       !
                        ;               +-------------------------------+
```

```
;NORMAL PSEUDO-TERMINAL IN PACKET DESCRIPTION GIVEN IN RESPONSE TO ABOVE PACKET

;BYTE OFFSET FROM
;START OF PACKET      +-----------------------------------+
;         0           !          REQUEST NUMBER           !    ECHOED FROM REQUEST PACKET
;                     +-----------------------------------+
;         2           !         DATA ARGUMENT #1          !    R3 CONTAINS THIS ADDRESS
;                     +-----------------------------------+
;         4           !         DATA ARGUMENT #2          !    ALL DATA ARGUMENTS ARE RETURNED
;                     +-----------------------------------+    CONTAINING ZEROS UNLESS
;         6           !         DATA ARGUMENT #3          !    SPECIFICALLY INDICATED BY
;                     +-----------------------------------+    RESPONSE ROUTINE.
;         8           !         DATA ARGUMENT #4          !
;                     +-----------------------------------+
;        10           !         DATA ARGUMENT #5          !
;                     +-----------------------------------+
;        12           !         DATA ARGUMENT #6          !
;                     +-----------------------------------+
;        14           !         DATA ARGUMENT #7          !
;                     +-----------------------------------+
;        16           !         DATA ARGUMENT #8          !
;                     +-----------------------------------+
;        18           !         DATA ARGUMENT #9          !
;                     +-----------------------------------+
;        20           !         DATA ARGUMENT #10         !
;                     +-----------------------------------+
;        22           !         DATA ARGUMENT #11         !
;                     +-----------------------------------+
;
;
;         .                         .
;
;         .                         .
;
;         .                         .
;
;                     +-----------------------------------+
;        68           !         DATA ARGUMENT #34         !
;                     +-----------------------------------+
```

```
                                ;T1MSIZ  -  DM REQUEST 0
                                ;
                                ;SET UP MEMORY FOR ADDRESS TESTING FROM UDA.
                                ;PLACE ADDRESS OF EACH LOCATION INTO EACH LOCATION IN FREE
                                ;MEMORY. RETURN FIRST LOCATION OF FREE MEMORY IN CMD.02 (LOW BITS)
                                ;AND CMD.03 (HIGH BITS). RETURN LAST LOCATION OF FREE MEMORY IN
                                ;CMD.04 AND CMD.05. ALSO RETURN FIRST EXISTANT LOCATION IN CMD.06
                                ;AND CMD.07; LAST EXISTANT LOCATION IN CMD.08 AND CMD.09.
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE PACKET DATA ADDRESS (POINTING TO MSG.02)
                                ;       R3 - COMMAND PACKET DATA ADDRESS (POINTING TO CMD.02)
                                ;OUTPUTS:
                                ;       COMMAND PACKET CONTAINING:
                                ;         (R3) LOW ADDRESS BITS OF FIRST WRITABLE ADDRESS
                                ;       2.(R3) HIGH ADDRESS BITS OF FIRST WRITABLE ADDRESS
                                ;       4.(R3) LOW ADDRESS BITS OF LAST WRITABLE ADDRESS
                                ;       6.(R3) HIGH ADDRESS BITS OF LAST WRITABLE ADDRESS
                                ;       8.(R3) LOW ADDRESS BITS OF FIRST READABLE ADDRESS
                                ;      10.(R3) HIGH ADDRESS BITS OF FIRST READABLE ADDRESS
                                ;      12.(R3) LOW ADDRESS BITS OF LAST READABLE ADDRESS
                                ;      14.(R3) HIGH ADDRESS BITS OF LAST READABLE ADDRESS
                                ;       Z SET

077710                          T1MSIZ:
077710  013701  065016                  MOV FFREE,R1                    ;GET FIRST ADDRESS OF FREE MEMORY
077714  013702  065020                  MOV FSIZE,R2                    ;GET SIZE

                                ;FILL MEMORY WITH ADDRESS PATTERN

077720  010111                  MEMFIL: MOV R1,(R1)                     ;WRITE DATA INTO LOCATION
077722  062701  000002                  ADD #2,R1                       ;INCREASE ADDRESS TO NEXT LOCATION
077726  005302                          DEC R2                          ;COUNT THE WORDS
077730  001373                          BNE MEMFIL                      ;FILL ALL WORDS

                                ;SEND LOCATION OF FREE MEMORY TO UDA

077732  013723  065016                  MOV FFREE,(R3)+                 ;LOAD FIRST ADDRESS OF FREE MEMORY
077736  005023                          CLR (R3)+                       ; HIGH ORDER BITS ARE ZERO
077740  013700  065020                  MOV FSIZE,R0                    ;GET SIZE OF FREE MEMORY
077744  006300                          ASL R0                          ;CONVERT TO BYTES
077746  063700  065016                  ADD FFREE,R0                    ;COMPUTE LAST LOCATION
077752  162700  000002                  SUB #2,R0
077756  010023                          MOV R0,(R3)+                    ;LOAD LAST LOCATION
077760  005023                          CLR (R3)+                       ;CLEAR HIGH ORDER BITS
```

GLOBAL SUBROUTINES SECTION

```
                                ;SEND LOCATION OF READABLE MEMORY
                                
077762  005023                  CLR (R3)+               ;SEND ZERO AS START OF READABLE MEMORY
077764  005023                  CLR (R3)+
077766  013700  002120          MOV L$HIMEM,R0          ;GET HIGH MEMORY ADDRESS
077772  005001                  CLR R1                  ;CLEAR HIGH BITS
077774  006300                  ASL R0                  ;SHIFT LEFT 6 PLACES
077776  006300                  ASL R0
100000  006300                  ASL R0
100002  006300                  ASL R0
100004  006300                  ASL R0
100006  006101                  ROL R1
100010  006300                  ASL R0
100012  006101                  ROL R1
100014  052700  000076          BIS #76,R0              ;SET LOW ORDER BITS
100020  010023                  MOV R0,(R3)+            ;PUT INTO BUFFER
100022  010123                  MOV R1,(R3)+
100024  000264                  SEZ
100026  000207                  RETURN
```

```
                                ;T2DLL - DM REQUEST 1
                                ;
                                ;PROVIDE DIAGNOSTIC TO DOWNLINE LOAD INTO DISK DRIVE.
                                ;
                                ;THE UDA MAY BE USED TO GET THE DIAGNOSTIC IF THE SYSTEM LOAD DEVICE
                                ;IS ON THE UDA. THIS ACTION WILL CAUSE A REINITIALIZATION OF THE UDA
                                ;AND THE RING STRUCTURE MOVED. SINCE THIS PROGRAM HAS NO WAY TO
                                ;DETERMINE IF THE UDA IS USED, IT WILL ALWAYS ASSUME IT IS USED AND
                                ;WILL INITIALIZE AND RELOAD THE DM PROGRAM AFTER READING THE
                                ;DIAGNOSTIC. THE OUTPUTS OF THIS ROUTINE ARE STORED AND SENT TO THE
                                ;DM PROGRAM IN THE UTOTST REQUEST.
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;          (R4) DRIVE NUMBER
                                ;        2.(R4) A VALUE THE DM PROGRAM WISHES RETURNED
                                ;        4.(R4) REGION TO WHICH PROGRAM IS TO BE LOADED IN DISK
                                ;        6.(R4) 2 WORD PROGRAM NAME IN RAD50
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       COMMAND PACKET COULD CONTAIN THE FOLLOWING:
                                ;          (R3) ONE IF PROGRAM PROVIDED, TWO IF PROGRAM NOT AVAILABLE
                                ;        2.(R3) DRIVE NUMBER
                                ;        4.(R3) COPY OF THE VALUE FROM DM PROGRAM
                                ;        6.(R3) REGION TO WHICH PROGRAM IS TO BE LOADED
                                ;        8.(R3) ADDRESS OF FIRST BYTE TO BE DOWNLINE LOADED
                                ;       10.(R3) HIGH ORDER BITS OF ADDRESS
                                ;       12.(R3) BYTE COUNT OF PROGRAM TO BE DOWNLINE LOADED
                                ;       Z SET
                                ;THIS PROGRAM WILL NOT SEND A COMMAND PACKET IN RESPONSE TO THIS REQUEST.
                                ;THE UDA WILL BE REINITIALIZED AND THE DM PROGRAM RELOADED. THEN THIS DATA
                                ;WILL BE APPENDED TO THE NEXT UTOTST REQUEST.

                                ;COPY REQUEST DATA TO STORAGE

100030                          T2DLL:
100030  005037  065262                  CLR DLL                 ;CLEAR CONTROL WORD
100034  012437  065264                  MOV (R4)+,DLLDR         ;DRIVE NUMBER
100040  012437  065266                  MOV (R4)+,DLLV          ;VALUE FROM DM
100044  012437  065270                  MOV (R4)+,DLLR          ;REGION
100050  012437  065300                  MOV (R4)+,DLLNAM        ;PROGRAM NAME
100054  012437  065302                  MOV (R4)+,DLLNAM+2      ;  (TWO WORDS)
```

GLOBAL SUBROUTINES SECTION

```
                                        ;RESET UDA AND READ DM PROGRAM

100060  005075  000000                  CLR @(R5)                   ;RESET THE UDA
100064  013737  065016  065272          MOV FFREE,DLLADR            ;GET ADDRESS WHERE PROGRAM
100072  005037  065274                  CLR DLLADR+2                ;  TO BE STORED
100076  013737  065020  065276          MOV FSIZE,DLLSIZ            ;SAVE CURRENT SIZE OF MEMORY
100104  004737  107004                  CALL RDDLL                  ;READ DLL PROGRAM FROM DATA FILE
100110  103002                          BCC 1$                      ;PROGRAM NOT FOUND IF CARRY SET
100112  005237  065262                  INC DLL                     ;RETURN 1 IF PROGRAM FOUND
100116  005237  065262          1$:     INC DLL                     ;RETURN 2 IF PROGRAM NOT FOUND
100122  013737  065276  065020          MOV DLLSIZ,FSIZE            ;COMPUTE SIZE OF DLL PROGRAM
100130  013737  065016  065276          MOV FFREE,DLLSIZ            ;  AND RESTORE ORIGINAL FFREE
100136  163737  065272  065276          SUB DLLADR,DLLSIZ           ;  AND FSIZE VALUES
100144  013737  065272  065016          MOV DLLADR,FFREE
100152  005726                          TST (SP)+                   ;POP RETURN ADDRESS OFF STACK
100154  012701  000001                  MOV #1,R1                   ;RUN THE DM PROGRAM AGAIN
100160  004737  077022                  CALL RUNDM
100164  001402                          BEQ 2$
100166  000137  077120                  JMP RESPDM
100172  000207                  2$:     RETURN
```

```
                                ;T2CMD - DM REQUEST 2
                                ;
                                ;GET MANUAL INTERVENTION COMMAND
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;          (R4) DRIVE NUMBER
                                ;        2.(R4) OPERATION CODE
                                ;               0 ON FIRST REQUEST FOR DRIVE. ECHO OF PREVIOUS RESPONSE ALL OTHER TIMES.
                                ;          IF OPERATION CODE = 2
                                ;        4.(R4) DATA BYTE READ (TO BE PRINTED)
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       COMMAND DATA FILLED WITH THE FOLLOWING:
                                ;          (R3) OPERATION CODE
                                ;               0 - EXIT
                                ;               1 - WRITE
                                ;               2 - READ
                                ;               3 - DIAGNOSE
                                ;          IF OPERATION CODE = 1, 2 OR 3
                                ;        2.(R3) REGION NUMBER
                                ;        4.(R3) OFFSET INTO REGION
                                ;          IF OPERATION CODE = 1
                                ;        6.(R3) DATA BYTE
                                ;       Z SET IF DATA RETURNED
                                ;       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
100174                          T2CMD:
100174  032737  000200  065000          BIT #SM.MAN,SFPTBL+SO.BIT       ;LOOK AT MANUAL INTERVENTION MODE
100202  001002                        BNE T2CMDM                      ;EXIT IF NOT WANTED
100204  000137  100626                JMP T2CMDX
100210                          T2CMDM: MANUAL                          ;MANUAL INTERVENTION ALLOWED?
100210  104450                                                                  TRAP    C$MANI
100212                                  BCOMPLETE T2CMD0                ;PRINT WARNING IF NOT
100212  103406                                                                  BCS     T2CMD0
100214                          T2CMD9: PNTF T2WARN
100214  004137  104212                                                          JSR R1,LPNTF
100220  066363                                                                  .WORD T2WARN
100222  000000                                                                  .WORD PNT.CT
100224  000137  100626                JMP T2CMDX
100230  012401                  T2CMD0: MOV (R4)+,R1                    ;GET DRIVE NUMBER
100232  012402                          MOV (R4)+,R2                    ;GET OPERATION CODE
100234  001022                          BNE T2CMD2                      ;BRANCH IF NOT ZERO
100236  004737  103132                  CALL GTDRVT                     ;GET DRIVE TABLE ADDRESS
100242  001401                          BEQ 1$                          ;CHECK IF DRIVE FOUND
100244  000207                          RETURN                          ;RETURN WITH Z CLEAR IF NOT
```

```
     1 100246                          1$:     PNTF T2CMS1,D.UNIT(R4),(R5),(R4)        ;PRINT DESCRIPTION
       100246  011446                                                                          MOV (R4),-(SP)
       100250  011546                                                                          MOV (R5),-(SP)
       100252  016446  000002                                                                 MOV D.UNIT(R4),-(SP)
       100256  004137  104212                                                                 JSR R1,LPNTF
       100262  066462                                                                        .WORD T2CMS1
       100264  000006                                                                        .WORD PNT.CT
     2 100266  005037  065246                  CLR T2WRR                     ;CLEAR ALL STORAGE WORDS
     3 100272  005037  065250                  CLR T2WRO
     4 100276  005037  065252                  CLR T2DR
```

```
      1 100302  022702  000002        T2CMD2: CMP #2,R2                ;SEE IF LAST OPERATION WAS READ
      2 100306  001027                        BNE T2CMDQ               ;BRANCH IF NOT TO QUESTION
      3 100310                                PRINT <#' >              ;PRINT ONE SPACE
        100310  112700  000040                                                 MOVB #' ,R0
        100314  004737  104040                                                 CALL CPNT
      4 100320  013701  065246                MOV T2WRR,R1             ;PRINT REGION
      5 100324  004737  103600                CALL T2PNTW
      6 100330  013701  065250                MOV T2WRO,R1             ;PRINT OFFSET
      7 100334  004737  103600                CALL T2PNTW
      8 100340                                PRINT #'/                ;PRINT A SLASH
        100340  112700  000057                                                 MOVB #'/,R0
        100344  004737  104040                                                 CALL CPNT
      9 100350  012401                        MOV (R4)+,R1             ;PRINT THE DATA
     10 100352  004737  103630                CALL T2PNTB
     11 100356                                PRINT #CR                ;END THE LINE
        100356  112700  000015                                                 MOVB #CR,R0
        100362  004737  104040                                                 CALL CPNT
```

```
                                ;NOW ASK FOR COMMAND INPUT
                                
100366                          T2CMDQ: GMANID T4OPT7,TEMP,A,-1,1,20.,NO        ;ASK FOR COMMAND
100366  104443                                                                          TRAP    C$GMAN
100370  000406                                                                          BR      10000$
100372  065102                                                                          .WORD   TEMP
100374  000142                                                                          .WORD   T$CODE
100376  065364                                                                          .WORD   T4OPT7
100400  177777                                                                          .WORD   -1
100402  000001                                                                          .WORD   T$LOLIM
100404  000024                                                                          .WORD   T$HILIM
100406                                                                          10000$:
100406  012701  065102                  MOV #TEMP,R1                    ;GET POINTER TO STRING
100412  112100                          MOVB (R1)+,R0                   ;GET COMMAND CHARACTER
100414  022700  000105                  CMP #'E,R0
100420  001415                          BEQ T2CMDV
100422  022700  000104                  CMP #'D,R0
100426  001016                          BNE T2CMD3
100430  012713  000003                  MOV #3,(R3)                     ;STORE DIAGNOSE OPERATION CODE
100434  004737  103712                  CALL T2GNUM                     ;GET REGION FROM COMMAND
100440  001402                          BEQ 1$
100442  010437  065252                  MOV R4,T2DR
100446  013763  065252  000002  1$:     MOV T2DR,2(R3)
100454  004737  103712          T2CMDV: CALL T2GNUM                     ;MAKE SURE AT END OF LINE
100460  001064                          BNE T2CMDE
100462  000461                          BR T2CMDX
```

```
                                        ;COMMAND MUST BE EITHER READ OR WRITE
                                        
100464  012713  000002          T2CMD3: MOV #2,(R3)                     ;CHECK IF READ
100470  022700  000122                  CMP #'R,R0
100474  001415                          BEQ T2CMDR
100476  022700  000127                  CMP #'W,R0                      ;CHECK IF WRITE
100502  001053                          BNE T2CMDE                      ;  IF NOT - ERROR
100504  012713  000001                  MOV #1,(R3)
100510  004737  103712                  CALL T2GNUM                     ;GET DATA BYTE
100514  001446                          BEQ T2CMDE                      ;ERROR IF NO DATA
100516  162700  000002                  SUB #2,R0
100522  003043                          BGT T2CMDE                      ;OR GREATER THAN TWO DIGITS
100524  010463  000006                  MOV R4,6(R3)                    ;STORE DATA BYTES IN BUFFER
100530  013763  065246  000002  T2CMDR: MOV T2WRR,2(R3)                 ;PUT REGION AND OFFSET
100536  013763  065250  000004          MOV T2WRO,4(R3)                 ;  INTO BUFFER
100544  021302                          CMP (R3),R2                     ;IF SO,
100546  001002                          BNE T2CMDN
100550  005263  000004                  INC 4(R3)                       ;  INCREMENT OFFSET
100554  004737  103712          T2CMDN: CALL T2GNUM
100560  001411                          BEQ T2CMDW
100562  010463  000002                  MOV R4,2(R3)
100566  005063  000004                  CLR 4(R3)
100572  004737  103712                  CALL T2GNUM
100576  001402                          BEQ T2CMDW
100600  010463  000004                  MOV R4,4(R3)
100604  004737  103712          T2CMDW: CALL T2GNUM
100610  001010                          BNE T2CMDE
100612  016337  000002  065246          MOV 2(R3),T2WRR                 ;SAVE REGION
100620  016337  000004  065250          MOV 4(R3),T2WRO                 ;SAVE OFFSET
100626  000264                  T2CMDX: SEZ
100630  000207                          RETURN
100632                          T2CMDE: PNTF T2CMS5                     ;REPORT ERROR MESSAGE
100632  004137  104212                                                          JSR R1,LPNTF
100636  067051                                                                  .WORD T2CMS5
100640  000000                                                                  .WORD PNT.CT
100642  000651                          BR T2CMDQ                       ;GO ASK AGAIN
```

```
                                        ;T4MPRM - DM REQUEST 3
                                        ;
                                        ;REQUEST FOR TEST 4 CONTENTS OF DATA PATTERN 16.
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R4 - MESSAGE DATA ADDRESS
                                        ;            (NO DATA)
                                        ;       R3 - COMMAND DATA ADDRESS
                                        ;OUTPUTS:
                                        ;       COMMAND DATA FILLED WITH THE FOLLOWING:
                                        ;          (R3) NUMBER OF WORDS IN DATA PATTERN 16
                                        ;         2.(R3) DATA IN PATTERN 16
                                        ;           !              "
                                        ;       32.(R3)            "
                                        ;       Z SET
                                        ;
100644  012701  000021          T4MPRM: MOV #17.,R1                     ;GET COUNT
100650  012702  065160                  MOV #PAT16C,R2                  ; AND ADDRESS OF PATTERN 16 PARAMETERS
100654  012223                  1$:     MOV (R2)+,(R3)+                 ;COPY THE DATA TO BUFFER
100656  005301                          DEC R1
100660  001375                          BNE 1$
100662  000264                          SEZ                             ;RETURN WITH Z SET
100664  000207                          RETURN
```

```
                        ;T4UPRM - DM REQUEST 4
                        ;
                        ;REQUEST FOR TEST 4 UNIT PARAMETERS
                        ;
                        ;INPUTS:
                        ;       R5 - CONTROLLER TABLE ADDRESS
                        ;       R4 - MESSAGE DATA ADDRESS
                        ;          (R4) DRIVE NUMBER
                        ;        2.(R4) DRIVE SERIAL NUMBER
                        ;          !
                        ;        6.(R4)
                        ;        8.(R4) HDA SERIAL NUMBER
                        ;          !
                        ;       14.(R4)
                        ;       R3 - COMMAND DATA ADDRESS
                        ;OUTPUTS:
                        ;       COMMAND DATA FILLED WITH THE FOLLOWING:
                        ;          (R3) PARAMETER BITS (1 FOR TRUE)
                        ;               BIT 14 - INITIAL WRITE
                        ;               BIT 13 - DIAGNOSTIC CYLINDERS
                        ;               BIT 12 - ECC CORRECTION
                        ;               BIT 11 - READ ONLY
                        ;               BIT 10 - WRITE ONLY
                        ;               BIT  9 - RETRIES
                        ;               BIT  8 - TRACK/GROUP AND CYLINDERS SPECIFIED
                        ;               BIT  7 - (NOT USED)
                        ;               BIT  6 - SEQUENTIAL SEEKS
                        ;               BIT  5 - BEGIN/END SETS SPECIFIED
                        ;               BIT  4 - TRACK SPECIFIED (0 - GROUPS SPECIFIED)
                        ;                        HAS MEANING ONLY WHEN BIT 5 IS ZERO
                        ;               BIT  3 - WRITE CHECKS ENABLED
                        ;               BIT  2 - WRITE CHECKS ALWAYS
                        ;               BIT  1 - DATA COMPARES ENABLED
                        ;               BIT  0 - DATA COMPARE ALWAYS
                        ;        2.(R3) DATA PATTERN NUMBER
                        ;       IF PARAMETER BIT 5 SET
                        ;        4.(R3) COUNT OF BEGIN/END SETS
                        ;        6.(R3) BEGIN BLOCK (2 WORDS) THEN END BLOCK (2 WORDS)
                        ;          !       1 TO 4 SETS
                        ;          !          OR
                        ;          !       IF COUNT OF BEGIN/END BLOCKS = 0
                        ;       36.(R3)    START CYLINDER (2 WORDS) THEN END CYLINDER (2 WORDS)
                        ;                  END CYLINDER A NEGATIVE VALUE IF TO TEST ENTIRE AREA
                        ;       IF PARAMETER BIT 5 CLEAR
                        ;        4.(R3) STARTING CYLINDER
                        ;        6.(R3)    (2 WORDS)
                        ;        8.(R3) ENDING CYLINDER (2 WORDS)
                        ;       10.(R3)    NEGATIVE FOR ALL CYLINDERS
                        ;       12.(R3) NUMBER OF TRACKS OR GROUPS SPECIFIED
                        ;       14.(R3) 1 TO 7 TRACK OR GROUP NUMBERS
                        ;          !       DETERMINED BY PARAMETER BIT 4
                        ;       26.(R3)
                        ;       Z SET IF DATA RETURNED
                        ;       Z CLEAR IF UNIT NUMBER NOT ON THIS CONTROLLER
```

```
                                T4UPRM: MOV (R4)+,R1                    ;GET DRIVE NUMBER
                                        MOV R4,R2                       ;SAVE DATA ADDRESS
                                        CALL GTDRVT                     ;GET DRIVE TABLE ADDRESS
                                        BNE T4UPRX                      ;CHECK IF DRIVE FOUND
                                        MOV (R2)+,D.SERN(R4)            ;COPY DRIVE SERIAL NUMBER TO DRIVE TABLE
                                        MOV (R2)+,D.SERN+2(R4)
                                        MOV (R2)+,D.SERN+4(R4)
                                        MOV D.PRM(R4),R1                ;GET PARAMETER BITS
                                        BIC #D.ZERO,R1                  ;CLEAR SOME BITS
                                        BIT #ISTRTH,IFLAGS              ;IF FIRST TIME TEST 4 BEING RUN
                                        BEQ 2$                          ;  AFTER A START COMMAND
                                        BIT #SM.IW,SFPTBL+SO.BIT        ;GET INITIAL WRITE BIT
                                        BEQ 2$
                                        BIS #D.IW,R1                    ;MOVE INTO PARAMETER BITS
                                2$:     MOV R1,(R3)+                    ;PUT INTO BUFFER
                                        MOV D.PAT(R4),(R3)+             ;PUT PATTERN NUMBER IN BUFFER
                                        BIT #D.BE,R1                    ;CHECK BEGIN/END PARAMETER BIT
                                        BEQ 10$                         ;BRANCH IF NOT SET

                                ;RETURN BEGIN/END SETS

                                        MOV #4*4+1,R1                   ;COUNT OF SETS TIMES WORDS PER SET PLUS COUNT WORD
                                        MOV R4,R2                       ;GET INDEX INTO DRIVE TABLE
                                        ADD #D.BEC,R2
                                1$:     MOV (R2)+,(R3)+                 ;TRANSFER THE BEGIN/END SETS
                                        DEC R1
                                        BNE 1$
                                        BR T4UPRX

                                10$:    BIT #D.CYL,D.PRM(R4)            ;LOOK AT D.CYL BIT
                                        BEQ 20$                         ;BRANCH IF NOT SET

                                ;RETURN TRACKS/GROUPS AND CYLINDERS

                                        TST D.BEC(R4)                   ;CHECK IF ANY TRACKS/GROUPS
                                        BEQ 25$                         ;BRANCH IF NONE
                                        MOV #4,R1                       ;COUNT OF CYLINDER WORDS
                                        MOV R4,R2
                                        ADD #D.BCYL,R2
                                11$:    MOV (R2)+,(R3)+                 ;CYLINDERS
                                        DEC R1
                                        BNE 11$
                                        MOV #8.,R1
                                        MOV R4,R2
                                        ADD #D.BEC,R2
                                12$:    MOV (R2)+,(R3)+                 ;TRACKS/GROUPS
                                        DEC R1
                                        BNE 12$
                                        BR T4UPRX
```

```
                                        ;RETURN CYLINDERS ONLY
                                        
101066  052763  000040  177774  25$:    BIS #D.BE,-4(R3)          ;SET D.BE FOR DM PROGRAM
101074  005023                          CLR (R3)+                 ;SEND ZERO BEGIN/END COUNT
101076  012701  000004                  MOV #4,R1
101102  010402                          MOV R4,R2
101104  062702  000154                  ADD #D.BCYL,R2
101110  012223                  26$:    MOV (R2)+,(R3)+           ;CYLINDERS
101112  005301                          DEC R1
101114  001375                          BNE 26$
101116  000412                          BR T4UPRX
                                        
                                        ;RETURN ENTIRE AREA
                                        
101120  052763  000040  177774  20$:    BIS #D.BE,-4(R3)          ;SET D.BE FOR DM PROGRAM
101126  005023                          CLR (R3)+                 ;BEGIN/END COUNT OF ZERO
101130  005023                          CLR (R3)+                 ;START CYLINDER OF ZERO
101132  005023                          CLR (R3)+
101134  005023                          CLR (R3)+                 ;END CYLINDER NEGATIVE
101136  012723  177777                  MOV #-1,(R3)+
101142  000264                          SEZ
101144  000207                  T4UPRX: RETURN
```

```
                                        ;T4BB1 - DM REQUEST 5
                                        ;
                                        ;REQUEST FOR FIRST 14 BAD BLOCKS
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R4 - MESSAGE DATA ADDRESS
                                        ;          (R4) DRIVE NUMBER
                                        ;       R3 - COMMAND DATA ADDRESS
                                        ;OUTPUTS:
                                        ;       COMMAND DATA FILLED WITH BAD BLOCKS
                                        ;          (R3) COUNT OF BAD BLOCKS
                                        ;           2.(R3) BAD BLOCK 1 (LOW)
                                        ;           4.(R3)             (HIGH)
                                        ;            :
                                        ;            :
                                        ;       56.(R3) BAD BLOCK 14 (LOW)
                                        ;       58.(R3)              (HIGH)
                                        ;       Z SET IF DATA RETURNED
                                        ;       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
                                        
101146  011401                  T4BB1:  MOV (R4),R1                     ;GET DRIVE NUMBER
101150  004737  103132                  CALL GTDRVT                     ;GET DRIVE TABLE ADDRESS
101154  001007                          BNE T4BB1E                      ;CHECK IF DRIVE FOUND
101156  062704  000010                  ADD #D.BB,R4                    ;INCREASE ADDRESS TO DATA TO COPY
101162  012701  000035                  MOV #<1+<14.*2>>,R1             ;GET COUNT OF WORDS
101166  012423                  1$:     MOV (R4)+,(R3)+                 ;COPY THE WORDS
101170  005301                          DEC R1
101172  001375                          BNE 1$
101174  000207                  T4BB1E: RETURN
```

```
                                        ;T4BB2 - DM REQUEST 6
                                        ;
                                        ;REQUEST LAST TWO BAD BLOCKS
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R4 - MESSAGE DATA ADDRESS
                                        ;          (R4) DRIVE NUMBER
                                        ;       R3 - COMMAND DATA ADDRESS
                                        ;OUTPUTS:
                                        ;       COMMAND DATA FILLED WITH BAD BLOCKS 15 AND 16
                                        ;       Z SET IF DATA RETURNED
                                        ;       Z CLEAR IF UNIT NUMBER NOT ON THIS CONTROLLER
                                        ;
101176  011401                  T4BB2:  MOV (R4),R1                     ;GET DRIVE NUMBER
101200  004737  103132                  CALL GTDRVT                     ;GET DRIVE TABLE ADDRESS
101204  001007                          BNE T4BB2E                      ;CHECK IF DRIVE FOUND
101206  062704  000102                  ADD #D.BB15,R4                  ;INCREASE ADDRESS TO DATA TO COPY
101212  012701  000004                  MOV #4,R1                       ;GET COUNT OF WORDS
101216  012423                  1$:     MOV (R4)+,(R3)+                 ;COPY THE WORDS
101220  005301                          DEC R1
101222  001375                          BNE 1$
101224  000207                  T4BB2E: RETURN
```

```
                                        ;T4SOFT - DM REQUEST 7
                                        ;
                                        ;ADD TO SOFT ERROR AND ECC COUNTS
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R4 - MESSAGE DATA ADDRESS
                                        ;          (R4) DRIVE NUMBER
                                        ;           2.(R4) VALUE TO ADD TO SOFT ERROR COUNT
                                        ;           4.(R4) VALUE TO ADD TO ECC COUNT
                                        ;       R3 - COMMAND DATA ADDRESS
                                        ;
101226  012401                  T4SOFT: MOV (R4)+,R1                    ;GET DRIVE NUMBER
101230  010402                          MOV R4,R2                       ;SAVE DATA ADDRESS
101232  004737  103132                  CALL GTDRVT                     ;GET DRIVE TABLE ADDRESS
101236  001005                          BNE 1$                          ;CHECK IF DRIVE FOUND
101240  062264  000172                  ADD (R2)+,D.SERR(R4)            ;ADD TO SOFT ERROR COUNT
101244  062264  000176                  ADD (R2)+,D.ECCC(R4)            ;ADD TO ECC COUNT
101250  000264                          SEZ                             ;EXIT
101252  000207                  1$:     RETURN
```

```
      1 101254                          T4SEEK:
      2                                 ;       DM REQUEST 8.
      3                                 ;
      4                                 ;RECORD 1000 SEEKS COMPLETED ON DRIVE
      5                                 ;
      6                                 ;INPUTS:
      7                                 ;       R5 - CONTROLLER TABLE ADDRESS
      8                                 ;       R4 - MESSAGE DATA ADDRESS
      9                                 ;          (R4) DRIVE NUMBER
     10                                 ;       R3 - COMMAND DATA ADDRESS
     11                                 ;
     12 101254  011401                  MOV     (R4),R1                 ; GET DRIVE NUMBER
     13 101256  004737  103132          CALL    GTDRVT                  ; GET DRIVE TABLE ADDRESS
     14 101262  001003                  BNE     SEKERE                  ; CHECK IF DRIVE FOUND
     15 101264  005264  000174          INC     D.SEEK(R4)              ; COUNT THE BITS TRANSFERRED
     16 101270  000264                  SEZ                             ; NORMAL RETURN
     17 101272  000207                  SEKERE: RETURN
```

```
                                ;T4MXFR - DM REQUEST 9.
                                ;
                                ;RECORD 1M BITS TRANSFERRED ON UNIT. COMPARE TO TRANSFER LIMIT AND
                                ;REPORT LIMIT REACHED.
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;          (R4) DRIVE NUMBER
                                ;         2.(R4) VALUE TO ADD TO READ COUNT
                                ;         4.(R4) VALUE TO ADD TO WRITE COUNT
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;          (R3) BIT 15 SET IF TRANSFER LIMIT REACHED
                                ;       MESSAGE PRINTED IF TRANSFER LIMIT REACHED
                                ;       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
                                
101274  010402                  T4MXFR: MOV R4,R2                       ;GET MESSAGE DATA ADDRESS
101276  011401                          MOV (R4),R1                     ;GET DRIVE NUMBER
101300  004737  103132                 CALL GTDRVT                     ;GET DRIVE TABLE ADDRESS
101304  001053                          BNE MXFERE                      ;CHECK IF DRIVE FOUND
101306  005764  000002                 TST D.UNIT(R4)                  ;SEE IF UNIT HAS BEEN DROPPED
101312  100003                          BPL 1$                          ;CONTINUE IF STILL TO BE TESTED
101314                                          ASSUME DT.AVL EQ BIT15
101314  052713  100000                 BIS #BIT15,(R3)                 ;TELL DM PROGRAM TO STOP TESTING THIS UNIT
101320  000444                          BR MXFERX                       ;  AND EXIT WITHOUT ADDING TO ADDING TO COUNTS
                                
101322                          1$:
101322  066264  000002  000166         ADD 2(R2),D.XFRR(R4)            ;ADD MEGABITS READ
101330  066264  000004  000164         ADD 4(R2),D.XFRW(R4)            ;ADD MEGABITS WRITTEN
101336  005737  064776                 TST SFPTBL+SO.XL                ;SEE IF LIMIT SPECIFIED
101342  001433                          BEQ MXFERX                      ;BRANCH IF NOT
101344  026437  000166  064776         CMP D.XFRR(R4),SFPTBL+SO.XL     ;CHECK IF LIMIT REACHED
101352  103427                          BLO MXFERX                      ;BRANCH IF LIMIT NOT REACHED
101354                                  RFLAGS R0                       ;CHECK FLAGS
101354  104421                                                                  TRAP    C$RFLA
101356  032700  000040                 BIT #IDU,R0                     ;SEE IF DROPPING UNITS IS INHIBITED
101362  001023                          BNE MXFERX
101364  052713  100000                 BIS #BIT15,(R3)                 ;SET DROP UNIT BIT
101370  042765  000010  000014         BIC #CT.MSG,C.FLG(R5)           ;CLEAR MESSAGE RECEIVED FLAG
101376                                  PNTX MESSG,D.UNIT(R4),(R5),(R4) ;PRINT TESTING DONE
101376  011446                                                                  MOV (R4),-(SP)
101400  011546                                                                  MOV (R5),-(SP)
101402  016446  000002                                                          MOV D.UNIT(R4),-(SP)
101406  004137  104232                                                          JSR R1,LPNTX
101412  066317                                                                  .WORD MESSG
101414  000006                                                                  .WORD PNT.CT
101416  004737  107614                 CALL RNTIME                     ;PRINT RUNTIME
101422                                  PNTX MXFERP
101422  004137  104232                                                          JSR R1,LPNTX
101426  065574                                                                  .WORD MXFERP
101430  000000                                                                  .WORD PNT.CT
101432  000264                  MXFERX: SEZ                             ;NORMAL RETURN
101434  000207                  MXFERE: RETURN
```

```
                                ;UTOTST - DM REQUEST 10
                                ;
                                ;TELL DM PROGRAM WHICH DRIVES ARE SELECTED FOR TESTING
                                ;AND CLEAR STATISTICS IN DRIVE TABLE
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;            (NO DATA)
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       COMMAND PACKET CONTAINING UP TO 8 DRIVE NUMBERS.
                                ;         LIST IS ENDED BY A WORD WITH BIT 15 SET.
                                ;         FOLLOWING LIST IS THE INFORMATION FROM T2DLL REQUEST IF APPLICABLE.
                                ;       D.XFRW, D.XFRR, D.HERR, D.SERR, D.SEEK AND D.ECC CLEARED IN DRIVE TABLE
                                ;       Z SET
                                ;
101436  010504                  UTOTST: MOV R5,R4                       ;GET ADDRESS OF CONTROLLER TABLE
101440  062704  000020                  ADD #C.DRO,R4                   ;BUMP TO DRIVE TABLE POINTERS
101444  012702  000010                  MOV #8.,R2                      ;GET COUNT OF PORTS
101450  012400                  UTOT1:  MOV (R4)+,R0                    ;SEE IF DRIVE TABLE POINTER EXISTS
101452  001415                          BEQ UTOT2                       ;BRANCH IF NOT
101454  005760  000002                  TST D.UNIT(R0)                  ;LOOK IF UNIT AVAILABLE FOR TESTING
101460  100410                          BMI UTOT1A
101462                                      ASSUME DT.AVL EQ BIT15
101462  011023                          MOV (R0),(R3)+                  ;LOAD DRIVE NUMBER FROM TABLE
101464  062700  000164                  ADD #D.XFRW,R0                  ;CLEAR STATISTICS IN DRIVE TABLE
101470  012701  000011                  MOV #D.SIZE-D.XFRW/2,R1
101474  005020                  1$:     CLR (R0)+
101476  005301                          DEC R1
101500  001375                          BNE 1$
101502  005302                  UTOT1A: DEC R2                          ;COUNT THE DRIVE TABLES
101504  001361                          BNE UTOT1                       ;REPEAT FOR EACH TABLE
101506  012723  100000          UTOT2:  MOV #BIT15,(R3)+                ;TERMINATE LIST
101512  013723  065262                  MOV DLL,(R3)+                   ;GET DLL CONTROL WORD
101516  001407                          BEQ UTOT4                       ; IF NON-ZERO
101520  012701  065264                  MOV #DLLDR,R1                   ;  TRANSFER ALL DLL WORDS INTO BUFFER
101524  012702  000020                  MOV #<DLLNAM+4-DLLDR>,R2
101530  012123                  UTOT3:  MOV (R1)+,(R3)+
101532  005302                          DEC R2
101534  001375                          BNE UTOT3
101536  000264                  UTOT4:  SEZ
101540  000207                          RETURN                          ;RETURN WITH Z SET
```

```
                                ;ERRMES  - DM REQUEST 11
                                ;
                                ;PRINT AN ERROR MESSAGE
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;            (R4)  ERROR PC IN DM PROGRAM
                                ;          2.(R4)  <15:14> ERROR TYPE
                                ;                  <13:0 > ERROR NUMBER
                                ;          4.(R4)  DRIVE NUMBER (-1 IF NOT GIVEN)
                                ;          6.(R4)  MESSAGE POINTER
                                ;          8.(R4)  OPTIONAL PARAMETERS FOR ERROR PRINT ROUTINE
                                ;         10.(R4)                  "
                                ;            !                     "
                                ;            !                     "
                                ;         58.(R4)                  "
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       COMMAND PACKET CONTAINING THE FOLLOWING:
                                ;           (R3) - BIT 15 SET IF FATAL ERROR TO INDICATE DRIVE SHOULD NO LONGER BE TESTED
                                ;       Z SET TO INDICATE DATA RETURNED
                                ;       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
101542                          ERRMES:
101542  005764  000002                TST 2(R4)                       ;CHECK IF FATAL ERROR
101546  100406                        BMI 5$                          ;BRANCH IF NOT
101550                          RFLAGS R0                               ;LOOK AT FLAGS
101550  104421                                                                  TRAP    C$RFLA
101552  032700  000040                BIT #IDU,R0                     ;SEE IF ALLOWED TO DROP UNITS
101556  001014                        BNE 6$                          ;BRANCH IF NOT
101560  052713  100000                BIS #BIT15,(R3)                 ;SET DROP DRIVE BIT
101564  016400  000002        5$:     MOV 2(R4),R0                    ;SEE IF SOFT ERROR
101570  005100                        COM R0
101572  032700  140000                BIT #140000,R0
101576  001004                        BNE 6$                          ;BRANCH IF NOT
101600  032737  000400  065000        BIT #SM.SSF,SO.BIT+SFPTBL       ;SEE IF SOFT ERRORS SUPPRESSED
101606  001063                        BNE     ERRMSX                  ;DON'T PRINT IF SO
101610                          6$:
101610  042765  000010  000014        BIC #CT.MSG,C.FLG(R5)           ;CLEAR MESSAGE RECEIVED FLAG
101616  022737  000004  065050        CMP #4,TNUM                     ;IF TEST # 4,
101624  001004                        BNE 7$
101626  032737  001000  065000        BIT #SM.LOG,SFPTBL+SO.BIT       ;  SEE IF LOG BEING USED
101634  001005                        BNE ERRMSL
101636  004737  104316        7$:     CALL PNTERR                     ;IF NOT, PRINT THE ERROR MESSAGE
101642  103045                        BCC ERRMSX                      ;IF DRIVE HASN'T BEEN DROPPED, PRINT
101644  000244                        CLZ                             ;ELSE RETURN
101646  000207                        RETURN
```

```
      1 101650  005737  065254           ERRMSL: TST LBUFS                       ;SEE IF LOG BUFFER ESTABLISHED
      2 101654  001016                           BNE 1$                          ;  LBUFS CONTAINS ADDRESS IF ESTABLISHED
      3 101656  013701  065034                   MOV     DMPROG,R1
      4 101662  005721                           TST     (R1)+                   ; LBUFS <- (DMPROG)+2
      5 101664  010137  065254                   MOV     R1,LBUFS
      6 101670  010137  065256                   MOV     R1,LBUFN
      7 101674  067701  163134                   ADD     @DMPROG,R1
      8 101700  005741                           TST     -(R1)                   ; LBUFE <- (LBUFS) + ((DMPROG)) - 2
      9 101702  010137  065260                   MOV     R1,LBUFE
     10 101706  005037  065046                   CLR     FNUM
     11 101712  013701  065256           1$:     MOV LBUFN,R1                    ;GET ADDRESS OF DATA STORAGE AREA
     12 101716  062737  000106  065256           ADD #HC.BSZ,LBUFN               ;ADD BYTES OF STORAGE NEEDED
     13 101724  023737  065256  065260           CMP LBUFN,LBUFE                 ;SEE IF ENOUGH ROOM
     14 101732  103007                           BHIS 3$                         ;  BRANCH IF NOT
     15 101734  010521                           MOV R5,(R1)+                    ;STORE CONTROLLER TABLE ADDRESS
     16 101736  012700  000042                   MOV #<HC.BSZ-2>/2,R0            ;GET COUNT OF REST OF DATA IN WORDS
     17 101742  012421                   2$:     MOV (R4)+,(R1)+                 ;STORE DATA
     18 101744  005300                           DEC R0
     19 101746  001375                           BNE 2$
     20 101750  000402                                   BR ERRMSX
     21 101752  010137  065256           3$:     MOV R1,LBUFN                    ;RESTORE OLD VALUE OF LBUFN
     22 101756  000264                   ERRMSX: SEZ
     23 101760  000207                           RETURN
```

```
                                ;ERRMC - DM REQUEST 12.
                                ;
                                ;REPORT AN ERROR MESSAGE IDENTICAL TO DM REQUEST ERRMES
                                ;THEN ADD ONE TO THE ERROR COUNT FOR THE DRIVE AND SEE IF
                                ;ERROR LIMIT REACHED.
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;          (R4)  ERROR PC IN DM PROGRAM
                                ;         2.(R4)  < 9:8 > ERROR TYPE
                                ;                 < 7:0 > ERROR NUMBER
                                ;         4.(R4)  DRIVE NUMBER (-1 IF NOT GIVEN)
                                ;         6.(R4)  <15:12> TYPE
                                ;                 <11:0 > MESSAGE POINTER
                                ;         8.(R4)  OPTIONAL PARAMETERS FOR ERROR PRINT ROUTINE
                                ;        10.(R4)                 "
                                ;           !                    "
                                ;           !                    "
                                ;        58.(R4)                 "
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       COMMAND PACKET CONTAINING THE FOLLOWING:
                                ;         (R3) BIT 15 SET IF ERROR COUNT REACHED
                                ;                 TO INDICATE DRIVE SHOULD NO LONGER BE TESTED.
                                ;       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
                                ;       Z SET TO INDICATE DATA RETURNED

101762                          ERRMC:  PUSH R4
101762  010446                                                          MOV R4,-(SP)
101764  004737  101542                 CALL    ERRMES                  ; CALL REQUEST ERRMES
101770                          POP R4
101770  012604                                                          MOV (SP)+,R4
101772  005713                  TST (R3)                                ;SEE IF UNIT ALREADY TO BE DROPPED
101774  100436                  BMI 3$                                  ;  IF SO, JUST EXIT NOW
101776  016401  000004          MOV     4(R4),R1                        ; GET DRIVE NUMBER
102002  016402  000002          MOV 2(R4),R2                            ;GET ERROR TYPE
102006  004737  103132          CALL    GTDRVT                          ; GET DRIVE TABLE
102012  001031                  BNE     5$                              ; EXIT IF NO TABLE FOR UNIT
102014  042702  037777          BIC #^C140000,R2
102020  022702  100000          CMP #100000,R2                          ;CHECK IF HARD ERROR
102024  001022                  BNE 3$                                  ;BRANCH IF NOT
102026  005264  000170          INC     D.HERR(R4)                      ; COUNT THE ERROR
102032  026437  000170  064774  CMP     D.HERR(R4),SFPTBL+SO.EL         ; CHECK IF AT LIMIT
102040  103414                  BLO     3$                              ; IF LIMIT REACHED, BRANCH
102042                          RFLAGS R0                               ;LOOK AT THE FLAGS
102042  104421                                                          TRAP    C$RFLA
102044  032700  000040          BIT #IDU,R0                             ;SEE IF DROPPING UNITS INHIBITED
102050  001010                  BNE 3$                                  ;BRANCH IF SO
```

```
      8 102052                                    PNTX ERRLIM,D.UNIT(R4)          ; PRINT LIMIT REACHED
        102052  016446  000002                                                        MOV D.UNIT(R4),-(SP)
        102056  004137  104232                                                      JSR R1,LPNTX
        102062  065651                                                              .WORD ERRLIM
        102064  000002                                                              .WORD PNT.CT
      9 102066  052713  100000                    BIS #BIT15,(R3)                 ;SET STOP TESTING BIT
     13 102072  000264                    3$:     SEZ                             ; SET Z FOR NORMAL RETURN
     14 102074  000207                            RETURN                          ; RETURN TO CALLING PROGRAM
     15
     16 102076  000244                    5$:     CLZ                             ; FLAG AS ERROR
     17 102100  000207                            RETURN                          ; RETURN TO CALLING PROGRAM
```

```
                                ;MESSAG - DM REQUEST 13.
                                ;
                                ;PRINT A MESSAGE WITH HEADER AS FOLLOWS:
                                ;  "UNIT XX UDA AT XXXXXX DRIVE XXX RUNTIME HH:MM:SS "
                                ;ENTIRE MESSAGE IS PRINTED WITH PRINTX CALLS.
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;         (R4) DRIVE NUMBER
                                ;        2.(R4) MESSAGE POINTER
                                ;        2.(R4) MESSAGE POINTER
                                ;        4.(R4) OPTIONAL MESSAGE PARAMETERS
                                ;          :
                                ;          :
                                ;       58.(R4) COMMAND DATA ADDRESS

102102  042765  000010  000014  MESSAG: BIC #CT.MSG,C.FLG(R5)          ;CLEAR MESSAGE RECEIVED FLAG
102110  012401                        MOV (R4)+,R1                    ;GET DRIVE NUMBER
102112                              PUSH R4                           ;SAVE DATA POINTER
102112  010446                                                                  MOV R4,-(SP)
102114  004737  103132              CALL GTDRVT                       ;GET DRIVE TABLE ADDRESS
102120  001033                      BNE 1$                            ;CHECK IF DRIVE FOUND
102122  005764  000002              TST D.UNIT(R4)                    ;IF UNIT DROPPED FROM TESTING
102126  100430                      BMI 1$                            ;  DON'T PRINT ANYTHING
102130                              PNTX MESSG,D.UNIT(R4),(R5),(R4) ;PRINT HEADER
102130  011446                                                                  MOV (R4),-(SP)
102132  011546                                                                  MOV (R5),-(SP)
102134  016446  000002                                                          MOV D.UNIT(R4),-(SP)
102140  004137  104232                                                          JSR R1,LPNTX
102144  066317                                                                  .WORD MESSG
102146  000006                                                                  .WORD PNT.CT
102150  004737  107614              CALL RNTIME                       ;GET RUNTIME PARAMETERS
102154                              POP R4
102154  012604                                                                  MOV (SP)+,R4
102156  012402                      MOV (R4)+,R2                      ;GET MESSAGE POINTER
102160  006302                      ASL R2                            ;DOUBLE TO MAKE BYTE OFFSET
102162  063702  065034              ADD DMPROG,R2                     ;ADD TO START OF MESSAGE STRINGS
102166  067702  162642              ADD @DMPROG,R2                    ;ADD SIZE OF MAIN PROGRAM
102172  105712                      TSTB (R2)                         ;CHECK FIRST BYTE
102174  001001                      BNE 2$                            ;IF ZERO
102176  005202                      INC R2                            ;  INCREMENT TO NEXT BYTE
102200  004737  102220      2$:     CALL OSTRNG                       ;OUTPUT ACCORDING TO STRING
102204  000264                      SEZ
102206  000207                      RETURN
102210                      1$:     POP R4
102210  012604                                                                  MOV (SP)+,R4
102212  000207                      RETURN
```

```
                                ;DONE - DM REQUEST 14
                                ;
                                ;MARK DM PROGRAM AS NO LONGER RUNNING
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;            (NO DATA)
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       Z CLEAR TO DROP UNIT FROM TESTING
                                
102214  000244                  DONE:   CLZ                             ;DROP UNIT FROM TESTING
102216  000207                          RETURN
```

```
                                ;OSTRNG
                                ;
                                ;OUTPUT A MESSAGE ACCORDING TO A FORMAT STRING
                                ;FORMAT OF THE ASCIZ STRING IS AS FOLLOWS:
                                ;
                                ;CHARACTERS ENCLOSED IN QUOTES ARE TO BE PRINTED AS THEY ARE.
                                ;
                                ;OTHERWISE CODE IS A SINGLE LETTER FOLLOWED BY AN OPTIONAL DECIMAL
                                ;NUMBER:
                                ;   ON - PRINT OCTAL NUMBER. N REPRESENTS SIZE OF BINARY NUMBER PASSED
                                ;        IN PARAMETER IN BITS. MAY BE IN RANGE 1 TO 32. IF N>16, TWO PARAMETER
                                ;        WORDS ARE USED, OTHERWISE ONLY ONE WORD. LEADING ZEROS ARE PRINTED.
                                ;        N IS ALWAYS SPECIFIED.
                                ;   DN - PRINT UNSIGNED DECIMAL NUMBER FROM N BIT PARAMETER. LEADING ZEROS
                                ;        ARE NOT PRINTED. A 16 BIT NUMBER EQUAL TO ZERO WILL PRINT "0".
                                ;   HN - PRINT HEX NUMBER FROM PARAMETER OF N BITS. IF N>16 TWO PARAMETERS
                                ;        ARE USED, OTHERWISE ONLY ONE PARAMETER. LEADING ZEROS ARE PRINTED.
                                ;   SN - PRINT N SPACES. N ASSUMED TO BE 1.
                                ;   NN - START NEW LINE (CR-LF SEQUENCE). N ASSUMED TO BE 1.
                                ;   AN - PRINT N ASCII CHARACTERS FROM PARAMETERS, N ASSUMED TO BE 1.
                                ;        N/2 PARAMETER WORDS USED.
                                ;   RN - EXECUTE ROUTINE #N. N MUST BE GIVEN AND DEFINED IN HOST PROGRAM.
                                ;
                                ;A NULL CHARACTER MEANS END OF MESSAGE. A NULL AS FIRST CHARACTER IN STRING
                                ;MUST BE IGNORED.
                                ;
                                ;INPUTS:
                                ;       R2 - ADDRESS OF START OF FORMAT STRING
                                ;       R4 - ADDRESS OF PARAMETERS
                                ;OUTPUTS:
                                ;       R2 AND R4 UPDATED TO END OF STRING AND PARAMETERS
                                
102220  112201          OSTRNG: MOVB (R2)+,R1           ;GET CONTROL CHARACTER
102222  001421                  BEQ OSTRE               ;EXIT IF NULL CHARACTER
102224  012700  102532          MOV #ERRC,R0            ;GET POINTER TO CHARACTER TABLE
102230  120110          NCONS:  CMPB R1,(R0)            ;COMPARE CHARACTER WITH TABLE ENTRY
102232  001407                  BEQ NCONF               ;BRANCH IF MATCH FOUND
102234  105720                  TSTB (R0)+              ;INCREMENT POINTER
102236  001374                  BNE NCONS               ;CONTINUE SEARCH IF NOT END OF TABLE
102240                          PNTF ERRME1             ;REPORT BAD CONTROL CHARACTER
102240  004137  104212                                                  JSR R1,LPNTF
102244  065505                                                          .WORD ERRME1
102246  000000                                                          .WORD PNT.CT
102250  000406                  BR OSTRE
102252  162700  102532  NCONF: SUB #ERRC,R0             ;GET INCREMENT INTO TABLE
102256  006300                  ASL R0                  ;DOUBLE TO WORD COUNT
102260  004770  102544          CALL @ERRD(R0)          ;DISPATCH TO PRINT ROUTINE
102264  000755                  BR OSTRNG               ;GET NEXT
102266  000207          OSTRE:  RETURN
```

```
                                        ;CONTROL CHARACTER WAS A QUOTE. PRINT ALL CHARACTERS TO THE NEXT QUOTE.

102270  112200                  CON.QU: MOVB (R2)+,R0           ;GET CHARACTER
102272  120027  000042                  CMPB R0,#''             ;CHECK IF ENDING QUOTE
102276  001403                          BEQ CON.QX              ;IF SO, GO GET NEXT CONTROL CHARACTER
102300                                  PRINT R0                ;PRINT THE CHARACTER
102300  004737  104040                                                  CALL CPNT
102304  000771                          BR CON.QU               ;CONTINUE PRINTING
102306  000207                  CON.QX: RETURN

                                        ;CONTROL CHARACTER WAS AN A. PRINT ASCII CHARACTERS FROM PARAMETERS.

102310  004737  103222          CON.A:  CALL GETCNT              ;GET COUNT OF CHARACTERS
102314                          CON.A1: PRINT (R4)+              ;PRINT THE CHARACTER
102314  112400                                                          MOVB (R4)+,R0
102316  004737  104040                                                  CALL CPNT
102322  005301                          DEC R1                  ;COUNT THE CHARACTERS
102324  001373                          BNE CON.A1              ;PRINT UNTIL COUNT REACHES ZERO
102326  032704  000001                  BIT #1,R4               ;CHECK IF R4 NOW ODD
102332  001401                          BEQ CON.A2
102334  005204                          INC R4                  ;IF SO, INCREMENT TO NEXT EVEN ADDRESS
102336  000207                  CON.A2: RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS A D. PRINT DECIMAL NUMBER.

102340  012701  000012          CON.D:  MOV #10.,R1              ;LOAD RADIX
102344  004737  103300                  CALL PNTNUM             ;PRINT NUMBER
102350  000207                          RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                        ;CONTROL CHARACTER WAS AN H. PRINT HEX NUMBER.

102352  012701  000020          CON.H:  MOV #16.,R1              ;LOAD RADIX
102356  004737  103300                  CALL PNTNUM             ;PRINT NUMBER
102362  000207                          RETURN                  ;NOW GET NEXT CONTROL CHARACTER
```

```
                                ;CONTROL CHARACTER WAS AN O. PRINT OCTAL NUMBER.

                        CON.O:  MOV #8.,R1              ;LOAD RADIX
                                CALL PNTNUM             ;PRINT NUMBER
                                RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                ;CONTROL CHARACTER WAS AN N. PRINT NEW LINE SEQUENCE.

                        CON.N:  CALL GETCNT             ;GET COUNT
                        CON.N1: PRINT #CR               ;PRINT NEW LINE SEQUENCE
                                                                        MOVB #CR,R0
                                                                        CALL CPNT
                                DEC R1                  ;COUNT THE SEQUENCES
                                BNE CON.N1
                                RETURN                  ;NOW GET NEXT CONTROL CHARACTER

                                ;CONTROL CHARACTER WAS AN R. CALL A PRE-PROGRAMMED ROUTINE.

                        CON.R:  CALL GETCNT             ;GET ROUTINE NUMBER
                                CMP R1,#ERRRSZ          ;CHECK IF DEFINED ROUTINE NUMBER
                                BHI CON.R1
                                ADD R1,R1               ;DOUBLE COUNT TO GET WORD INDEX
                                CALL @ERRRTB-2(R1)      ;CALL ROUTINE
                                RETURN                  ;NOW GET NEXT CONTROL CHARACTER
                        CON.R1: PNTF ERRME1             ;REPORT BAD MESSAGE STRING
                                                                        JSR R1,LPNTF
                                                                        .WORD ERRME1
                                                                        .WORD PNT.CT
                                POP R1                  ;FIX THE STACK
                                                                        MOV (SP)+,R1
                                RETURN

                                ;CONTROL CHARACTER WAS AN S. PRINT SPACES.

                        CON.S:  CALL GETCNT             ;GET COUNT
                        CON.S1: PRINT <#' >             ;PRINT A SPACE
                                                                        MOVB #' ,R0
                                                                        CALL CPNT
                                DEC R1                  ;COUNT THE SPACES
                                BNE CON.S1
                                RETURN                  ;NOW GET NEXT CONTROL CHARACTER
```

```
                                ;ERROR ROUTINE DISPATCH TABLE

102500  102564          ERRRTB: .WORD CALR1                     ;CALL ALTERNATE PRINT STRING IN DM MEMORY IMAGE
102502  102612                  .WORD CALR2                     ;PRINT AN SDI DIAGNOSE RESPONSE
102504  102710                  .WORD CALR3                     ;DECIDE WHETHER TO PRINT RBN
102506  102724                  .WORD CALR4                     ;PRINT BASIC LINE WITHOUT UDA ADDRESS
102510  103000                  .WORD CALR5                     ;PRINT BASIC LINE WITH UDA ADDRESS
102512  103056                  .WORD CALR6                     ;CALL ALTERNATE PRINT STRING IN PDP-11 MEMORY
102514  103072                  .WORD CALR7                     ;PRINT  "REPLACE UDA MODULE M7161"
102516  103110                  .WORD CALR8                     ;PRINT  "  UDASA CONTAINS  XXXXXX"
102520  103126                  .WORD CALR9                     ;REPRINT LAST NUMBER
        000011                  ERRRSZ=<.-ERRRTB>/2

102522                  TNAMES:
102522  067274                  .WORD BASN1
102524  067320                  .WORD BASN2
102526  067340                  .WORD BASN3
102530  067360                  .WORD BASN4

                                ;BUILD TWO TABLES
                                ;       FIRST CONTAINING CONTROL CHARACTERS
                                ;       SECOND CONTAINING ROUTINE ADDRESSES

                                .MACRO BUILD
                                        ENTRY '',CON.QU
                                        ENTRY A,CON.A
                                        ENTRY D,CON.D
                                        ENTRY H,CON.H
                                        ENTRY O,CON.O
                                        ENTRY N,CON.N
                                        ENTRY R,CON.R
                                        ENTRY S,CON.S
                                .ENDM
```

```
                                ;HERE IS FIRST TABLE

                                .MACRO ENTRY ARG1,ARG2
                                        .LIST
                                        .BYTE ''ARG1
                                        .NLIST
                                .ENDM

102532                          ERRC:   BUILD
102532  042                             .BYTE '''
102533  101                             .BYTE 'A
102534  104                             .BYTE 'D
102535  110                             .BYTE 'H
102536  117                             .BYTE 'O
102537  116                             .BYTE 'N
102540  122                             .BYTE 'R
102541  123                             .BYTE 'S
102542  000                             .BYTE 0                 ;FOLLOW WITH A NULL BYTE
                                        .EVEN

                                ;HERE IS SECOND TABLE

                                .MACRO ENTRY ARG1,ARG2
                                        .LIST
                                        .WORD ARG2
                                        .NLIST
                                .ENDM

102544                          ERRD:   BUILD
102544  102270                          .WORD CON.QU
102546  102310                          .WORD CON.A
102550  102340                          .WORD CON.D
102552  102352                          .WORD CON.H
102554  102364                          .WORD CON.O
102556  102376                          .WORD CON.N
102560  102420                          .WORD CON.R
102562  102456                          .WORD CON.S
```

```
                                ;PRE-PROGRAMMED ROUTINE 1
                                ;CALL ALTERNATE PRINT STRING IN DM PROGRAM IMAGE

102564                          CALR1:  PUSH R2                 ;SAVE CURRENT STRING POINTER
102564  010246                                                          MOV R2,-(SP)
102566  012402                  MOV (R4)+,R2            ;GET NEW STRING POINTER
102570  006302                  ASL R2                  ;DOUBLE FOR WORD COUNT
102572  063702  065034          ADD DMPROG,R2           ;ADD START OF STRING STORAGE
102576  067702  162232          ADD @DMPROG,R2          ;ADD SIZE OF MAIN PROGRAM
102602  004737  102220          CALL OSTRNG             ;OUTPUT USING THIS STRING
102606                          POP R2                  ;GET OLD POINTER BACK
102606  012602                                                          MOV (SP)+,R2
102610  000207                  RETURN                  ;NOW CONTINUE THE OLD STRING
```

```
                                ;PRE-PROGRAMMED ROUTINE 2
                                ;PRINT AN SDI DIAGNOSE RESPONSE

102612                  CALR2:  PUSH R2                 ;SAVE CURRENT STRING POINTER
102612  010246                                                  MOV R2,-(SP)
102614  012402                  MOV (R4)+,R2            ;GET COUNTS
102616                          PUSH R2                 ;SAVE COUNTS
102616  010246                                                  MOV R2,-(SP)
102620  042702  177400          BIC #177400,R2          ;GET BINARY COUNT
102624  001414                  BEQ 2$                  ;BYPASS BINARY IF COUNT IS ZERO
102626  012700  000020  1$:     MOV #16.,R0             ;RADIX IS HEX
102632  012701  000040          MOV #32.,R1             ;32 BIT NUMBERS
102636  004737  103306          CALL PNTNUS             ;PRINT THE NUMBER
102642                          PRINT #CR               ;GO TO NEW LINE
102642  112700  000015                                          MOVB #CR,R0
102646  004737  104040                                          CALL CPNT
102652  005302                  DEC R2
102654  001364                  BNE 1$
102656                  2$:     POP R1                  ;GET ASCII COUNT
102656  012601                                                  MOV (SP)+,R1
102660  000301                  SWAB R1                 ;GET ASCII COUNT
102662  042701  177400          BIC #177400,R1
102666  001406                  BEQ 3$                  ;BYPASS IS COUNT IS ZERO
102670  004737  102314          CALL CON.A1             ;PRINT THE ASCII
102674                          PRINT #CR               ;GO TO NEW LINE
102674  112700  000015                                          MOVB #CR,R0
102700  004737  104040                                          CALL CPNT
102704                  3$:     POP R2                  ;RESTORE STRING POINTER
102704  012602                                                  MOV (SP)+,R2
102706  000207                  RETURN
```

```
                                ;PRE-PROGRAMMED ROUTINE 3
                                ;DECIDE WHETHER TO PRINT RBN
                                ;
                                ;FOUR PARAMETERS ARE PROVIDED FOR THIS ROUTINE. THE FIRST PARAMETER
                                ;SHOULD BE CHECKED TO SEE IF BIT 7 IS SET:
                                ;  IF SET - TURN INTO A CALL TO ROUTINE 1 (WHICH WILL USE OTHER 3 PARAMETERS)
                                ;  IF CLEAR - SKIP OVER NEXT 3 PARAMETERS AND END ROUTINE
                                
102710  032724  000200        CALR3:  BIT #BIT7,(R4)+                 ;CHECK BIT 7 IN FIRST PARAMETER WORD
102714  001323                         BNE CALR1                       ;IF SET, TURN INTO A CALR1
102716  062704  000006                 ADD #6,R4                       ;ELSE, SKIP OVER NEXT 3 PARAMETERS
102722  000207                         RETURN
```

```
                                ;PRE-PROGRAMMED ROUTINE 4
                                ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITHOUT UDA ADDRESS
                                ;THEN SWITCH TO EXTENDED FORMAT

102724                          CALR4:  PNTB BASLN,#BASNO,#BAS,#BAS,#BAS
102724  012746  067453                                                    MOV #BAS,-(SP)
102730  012746  067453                                                    MOV #BAS,-(SP)
102734  012746  067453                                                    MOV #BAS,-(SP)
102740  012746  067255                                                    MOV #BASNO,-(SP)
102744  004137  104222                                                    JSR R1,LPNTB
102750  067454                                                            .WORD BASLN
102752  000010                                                            .WORD PNT.CT
102754  004737  107614                  CALL RNTIME
102760                                  PRINT #CR
102760  112700  000015                                                    MOVB #CR,R0
102764  004737  104040                                                    CALL CPNT
102770  012737  104140  065012          MOV #PX,PTYPE
102776  000207                          RETURN
```

```
      1                                 ;PRE-PROGRAMMED ROUTINE 5
      2                                 ;PRINT BASIC LINE FOR HOST PROGRAM ERROR WITH UDA ADDRESS
      3                                 ;THEN SWITCH TO EXTENDED FORMAT
      4
      5 103000                          CALR5:  PNTB BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
        103000  012746  067453                                                          MOV #BAS,-(SP)
        103004  012746  067453                                                          MOV #BAS,-(SP)
        103010  011546                                                                  MOV (R5),-(SP)
        103012  012746  067417                                                          MOV #BASL2,-(SP)
        103016  012746  067255                                                          MOV #BASNO,-(SP)
        103022  004137  104222                                                          JSR R1,LPNTB
        103026  067454                                                                  .WORD BASLN
        103030  000012                                                                  .WORD PNT.CT
      6 103032  004737  107614                  CALL RNTIME
      7 103036                                  PRINT #CR
        103036  112700  000015                                                        MOVB #CR,R0
        103042  004737  104040                                                        CALL CPNT
      8 103046  012737  104140  065012         MOV #PX,PTYPE
      9 103054  000207                          RETURN
```

```
                                ;PRE-PROGRAMMED ROUTINE 6
                                ;CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
                                ;
103056                          CALR6:  PUSH R2                 ;SAVE CURRENT STRING POINTER
103056  010246                                                          MOV R2,-(SP)
103060  012402                          MOV (R4)+,R2            ;GET NEW STRING POINTER
103062  004737  102220                  CALL OSTRNG             ;OUTPUT USING THIS STRING
103066                                  POP R2                  ;GET OLD POINTER BACK
103066  012602                                                          MOV (SP)+,R2
103070  000207                          RETURN                  ;NOW CONTINUE THE OLD STRING
```

```
      1                                 ;PRE-PROGRAMMED ROUTINE 7
      2                                 ;PRINT ''REPLACE UDA MODULE M7161''
      3
      4 103072                          CALR7:  PUSH R2
        103072  010246                                                    MOV R2,-(SP)
      5 103074  012702  074555                  MOV #XFRU,R2
      6 103100  004737  102220                  CALL OSTRNG
      7 103104                                  POP R2
        103104  012602                                                    MOV (SP)+,R2
      8 103106  000207                          RETURN
```

```
                                ;PRE-PROGRAMMED ROUTINE 8
                                ;PRINT   ''  UDASA CONTAINS  XXXXXX''

103110                          CALR8:  PUSH R2
103110  010246                                                  MOV R2,-(SP)
103112  012702  074524                 MOV #XSA,R2
103116  004737  102220                 CALL OSTRNG
103122                                 POP R2
103122  012602                                                  MOV (SP)+,R2
103124  000207                         RETURN
```

```
                                ;       REPRINT LAST NUMBER
                                ;               R4 -> TABLE
103126  005744                  CALR9:  TST     -(R4)
103130  000207                          RETURN
```

```
                                        ;GTDRVT
                                        ;
                                        ;GET DRIVE TABLE POINTER
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;       R1 - DRIVE NUMBER
                                        ;OUTPUTS:
                                        ;       R4 - DRIVE TABLE ADDRESS
                                        ;       L$LUN - LOADED WITH UNIT NUMBER OF DRIVE
                                        ;       Z CLEAR IF DRIVE TABLE NOT FOUND AFTER ERROR PRINTED

103132                                  GTDRVT: PUSH R2
103132  010246                                                                    MOV R2,-(SP)
103134  010504                                  MOV R5,R4               ;GET CONTROLLER TABLE ADRESS
103136  062704  000020                          ADD #C.DRO,R4           ;ADD OFFSET TO DRIVE TABLE ADDRESS
103142  012702  000010                          MOV #8.,R2              ;GET COUNT OF DRIVES
103146  005714                          1$:     TST (R4)                ;CHECK IF AN ADDRESS HERE
103150  001406                                  BEQ 3$
103152  027401  000000                          CMP @(R4),R1            ;COMPARE DRIVE NUMBERS
103156  001412                                  BEQ 10$                 ;BRANCH IF A MATCH
103160  005724                          2$:     TST (R4)+               ;BUMP ADDRESS
103162  005302                                  DEC R2
103164  001370                                  BNE 1$                  ;LOOK AT ALL OF THEM
103166                                  3$:     ERRDF 35,,ERR035        ;UNIT NUMBER NOT FOUND
103166  104455                                                                  TRAP    C$ERDF
103170  000043                                                                  .WORD   35
103172  000000                                                                  .WORD   0
103174  075412                                                                  .WORD   ERR035
103176                                          POP R2
103176  012602                                                                  MOV (SP)+,R2
103200  000244                                  CLZ                     ;CLEAR Z AS ERROR FLAG
103202  000207                                  RETURN
103204  011404                          10$:    MOV (R4),R4             ;GET ADDRESS OF TABLE
103206  116437  000002  002074                  MOVB D.UNIT(R4),L$LUN   ;GET UNIT NUMBER
103214                                          POP R2
103214  012602                                                                  MOV (SP)+,R2
103216  000264                                  SEZ                     ;SET Z FLAG
103220  000207                                  RETURN
```

```
                                ;GETCNT
                                ;
                                ;GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
                                ;NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
                                ;DEFAULT OF 1.
                                ;
                                ;INPUTS:
                                ;       R2 - POINTER TO ASCII STRING
                                ;OUTPUTS:
                                ;       R1 - NUMBER READ OR A ONE
                                ;       R2 - POINTING TO CHARACTER AFTER NUMBER

103222                          GETCNT: PUSH R0
103222  010046                                                          MOV R0,-(SP)
103224  005001                          CLR R1                  ;START WITH ZERO COUNT
103226  121227  000060        GETCNX: CMPB (R2),#'0           ;CHECK IF CHARACTER A DIGIT
103232  103415                          BLO GETCDN              ;BRANCH IF LOWER THAN ZERO
103234  121227  000071                  CMPB (R2),#'9
103240  101012                          BHI GETCDN              ;BRANCH IF HIGHER THAN NINE
103242  006301                          ASL R1                  ;MULTIPLY NUMBER BY 10
103244  010100                          MOV R1,R0               ; SAVE 2N
103246  006301                          ASL R1                  ; COMPUTE 4N
103250  006301                          ASL R1                  ; COMPUTE 8N
103252  060001                          ADD R0,R1               ; 8N + 2N = 10N
103254  112200                          MOVB (R2)+,R0           ;GET DIGIT FROM STING
103256  162700  000060                  SUB #'0,R0              ;GET RID OF ASCII
103262  060001                          ADD R0,R1               ;ADD TO NUMBER
103264  000760                          BR GETCNX               ;GO TO NEXT CHARACTER
103266  005701                  GETCDN: TST R1                  ;CHECK IF NUMBER IS ZERO
103270  001001                          BNE GETCXX              ;IF ZERO, CHANGE
103272  005201                          INC R1                  ; TO DEFAULT OF ONE
103274                          GETCXX: POP R0
103274  012600                                                          MOV (SP)+,R0
103276  000207                          RETURN
```

```
                                ;PNTNUM
                                ;
                                ;PRINT A NUMBER
                                ;
                                ;INPUTS:
                                ;       R1 - RADIX OF NUMBER
                                ;       R2 - ASCII STRING TO COUNT OF BITS IN NUMBER
                                ;       R4 - POINTER TO NUMBER (LOW WORD)
                                ;OUTPUTS:
                                ;       NUMBER IS PRINTED. LEADING ZEROS ARE PRINTED EXCEPT FOR
                                ;            DECIMAL NUMBERS.
                                ;       R0 - CONTENTS DESTROYED
                                ;
103300  010100                  PNTNUM: MOV R1,R0                       ;SAVE RADIX
103302  004737  103222                  CALL GETCNT                     ;GET COUNT OF BITS
103306                          PNTNUS: PUSH <R2,R3,R5>
103306  010246                                                                  MOV R2,-(SP)
103310  010346                                                                  MOV R3,-(SP)
103312  010546                                                                  MOV R5,-(SP)
103314  012403                          MOV (R4)+,R3                    ;GET ONE PARAMETER WORD
103316  005005                          CLR R5                          ;CLEAR STORAGE FOR OTHER
103320  020127  000020                  CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
103324  003401                          BLE 1$
103326  012405                          MOV (R4)+,R5                    ;YES, GET SECOND PARAMETER WORD
103330                          1$:     PUSH R4
103330  010446                                                                  MOV R4,-(SP)
103332  010504                          MOV R5,R4                       ;PUT HIGH WORD IN R4
103334  012702  000020                  MOV #16.,R2                     ;COMPUTE BITS NOT WANTED
103340  160102                          SUB R1,R2                       ;BY SUBTRACTING BITS TO USE
103342  002002                          BGE 2$                          ;FROM 16.
103344  062702  000020                  ADD #16.,R2                     ;IF NEGATIVE, ADD 16 FOR FIRST WORD
103350  001414                  2$:     BEQ 6$                          ;IF ZERO, NO BITS NEED BE CLEARED
103352  012705  100000                  MOV #BIT15,R5                   ;START MASK WITH SIGN BIT SET
103356  005302                  3$:     DEC R2                          ;COUNT BITS IN MASK
103360  001402                          BEQ 4$
103362  006205                          ASR R5                          ;SHIFT MORE BITS TO RIGHT
103364  000774                          BR 3$
103366  020127  000020          4$:     CMP R1,#16.                     ;MORE THAN 16 BITS IN NUMBER?
103372  003402                          BLE 5$
103374  040504                          BIC R5,R4                       ;YES, CLEAR IN HIGH WORD
103376  000401                          BR 6$
103400  040503                  5$:     BIC R5,R3                       ;NO, CLEAR IN LOW WORD
103402  004737  103542          6$:     CALL DIVIDE                     ;DIVIDE BY RADIX IN R0
103406                                  PUSH R5                         ;PUSH REMAINDER ON STACK
103406  010546                                                                  MOV R5,-(SP)
103410  005202                          INC R2                          ;COUNT DIGITS ON STACK
103412  005703                          TST R3                          ;CHECK IF QUOTIENT IS ZERO
103414  001372                          BNE 6$
103416  005704                          TST R4
103420  001370                          BNE 6$
```

GLOBAL SUBROUTINES SECTION

```
    1 103422  020027  000012                CMP R0,#10.             ;IF RADIX IS DECIMAL
    2 103426  001423                        BEQ 10$                 ; JUST GO PRINT DIGITS ON STACK
    3 103430  010103                        MOV R1,R3               ;OTHERWISE COMPUTE NUMBER OF LEADING ZEROS
    4 103432  162700  000014                SUB #12.,R0             ;DIVIDEND IS BITS IN NUMBER
    5 103436  003002                        BGT 7$                  ;DIVISOR IS BITS PER DIGIT PRINTED
    6 103440  012700  000003                MOV #3,R0               ;   (3 OR 4)
    7 103444  004737  103542        7$:     CALL DIVIDE
    8 103450  005705                        TST R5                  ;IF REMAINDER NOT ZERO
    9 103452  001401                        BEQ 8$                  ;INCREMENT QUOTIENT
   10 103454  005203                        INC R3
   11 103456  160203                8$:     SUB R2,R3               ;SUBTRACT DIGITS ON STACK
   12 103460  001406                        BEQ 10$                 ;NO LEADING ZEROS IF ZERO
   13 103462                        9$:     PRINT #'0               ;PRINT A ZERO
      103462  112700  000060                                                 MOVB #'0,R0
      103466  004737  104040                                                 CALL CPNT
   14 103472  005303                        DEC R3
   15 103474  001372                        BNE 9$                  ;REPEAT UNTIL COUNT REACHES ZERO
   16
   17 103476                        10$:    POP R5                  ;GET CHACACTER FROM STACK
      103476  012605                                                         MOV (SP)+,R5
   18 103500  062705  000060                ADD #'0,R5              ;CNVERT TO ASCII DIGIT
   19 103504  020527  000071                CMP R5,#'9              ;IF GREATER THAN A 9
   20 103510  003402                        BLE 11$                 ; CONVERT TO A OR HIGHER
   21 103512  062705  000007                ADD #<'A-'9-1>,R5       ; FOR HEX DIGIT
   22 103516                        11$:    PRINT R5                ;PRINT THE CHARACTER
      103516  110500                                                         MOVB R5,R0
      103520  004737  104040                                                 CALL CPNT
   23 103524  005302                        DEC R2                  ;REPEAT FOR ALL DIGITS
   24 103526  001363                        BNE 10$                 ; ON STACK
   25 103530                                POP <R4,R5,R3,R2>
      103530  012604                                                         MOV (SP)+,R4
      103532  012605                                                         MOV (SP)+,R5
      103534  012603                                                         MOV (SP)+,R3
      103536  012602                                                         MOV (SP)+,R2
   26 103540  000207                RETURN
```

```
                                ;DIVIDE
                                ;
                                ;DIVIDE A 32 BIT UNSIGNED NUMBER BY A 16 BIT UNSIGNED NUMBER.
                                ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
                                ;WILL NOT CHECK FOR DIVIDE BY ZERO.
                                ;
                                ;INPUTS:
                                ;       R3 - LOW 16 BITS OF DIVIDEND
                                ;       R4 - HIGH 16 BITS OF DIVIDEND
                                ;       R0 - DIVISOR
                                ;OUTPUTS:
                                ;       R3 - LOW 16 BITS OF QUOTIENT
                                ;       R4 - HIGH 16 BITS OF QUOTIENT
                                ;       R5 - REMAINDER
                                
103542                          DIVIDE: PUSH R2                                 MOV R2,-(SP)
103542  010246
103544  012702  000040                  MOV #32.,R2             ;SET UP SHIFT COUNT
103550  005005                          CLR R5                  ;START WITH ZERO REMAINDER
103552  006303                  1$:     ASL R3                  ;SHIFT LEFT INTO R5
103554  006104                          ROL R4
103556  006105                          ROL R5
103560  020005                          CMP R0,R5               ;WILL DIVISOR GO INTO REMAINDER
103562  101002                          BHI 2$                  ;ONLY SUBTRACT IF IT WILL
103564  160005                          SUB R0,R5               ;SUBTRACT DIVISOR
103566  005203                          INC R3                  ;PUT A ONE INTO QUOTIENT
103570  005302                  2$:     DEC R2                  ;COUNT THE SHIFTS
103572  001367                          BNE 1$
103574                                  POP R2
103574  012602                                                  MOV (SP)+,R2
103576  000207                          RETURN
```

```
                                        ;PRINT HEX NUMBERS WITH LEADING SPACE
                                        
103600                                  T2PNTW: PRINT <#' >                     ;PRINT A SPACE
103600  112700  000040                                                                  MOVB #' ,R0
103604  004737  104040                                                                  CALL CPNT
103610                                          PUSH R1                                 MOV R1,-(SP)
103610  010146
103612  000301                                  SWAB R1
103614  004737  103640                         CALL T2PNT              ;PRINT HIGH TWO DIGITS
103620                                          POP R1                                  MOV (SP)+,R1
103620  012601
103622  004737  103640                         CALL T2PNT              ;PRINT LOW TWO DIGITS
103626  000207                                  RETURN
                                        
103630                                  T2PNTB: PRINT <#' >                     ;PRINT A SPACE
103630  112700  000040                                                                  MOVB #' ,R0
103634  004737  104040                                                                  CALL CPNT
                                        
                                        ;PRINT TWO HEX DIGITS FROM NUMBER IN R1
                                        
103640                                  T2PNT:  PUSH R1                         ;SAVE NUMBER
103640  010146                                                                          MOV R1,-(SP)
103642  006001                                  ROR R1                          ;SHIFT TO GET HIGH DIGIT
103644  006001                                  ROR R1
103646  006001                                  ROR R1
103650  006001                                  ROR R1
103652  004737  103660                         CALL T2PNTO                     ;PRINT TWO DIGITS
103656                                          POP R1                          ;GET LOW DIGIT AGAIN
103656  012601                                                                          MOV (SP)+,R1
103660  042701  177760          T2PNTO: BIC #^C17,R1                    ;CLEAR OTHER BITS
103664  062701  000060                  ADD #'0,R1                      ;CONVERT TO ASCII CHARACTER
103670  020127  000071                  CMP R1,#'9                      ;IF GREATER THAN A 9
103674  003402                          BLE T2PNTD                      ;  CONVERT TO A OR HIGHER
103676  062701  000007                  ADD #<'A-'9-1>,R1               ;  FOR HEX DIGIT
103702                          T2PNTD: PRINT R1                        ;PRINT THE DIGIT
103702  110100                                                                  MOVB R1,R0
103704  004737  104040                                                          CALL CPNT
103710  000207                                  RETURN
```

```
                                ;T2GNUM
                                ;
                                ;GET A HEX DIGIT FROM AN ASCII INPUT STRING
                                ;
                                ;INPUTS:
                                ;       R1 - STRING POINTER
                                ;OUTPUTS:
                                ;       R4 - NUMBER
                                ;       R1 - UPDATED STRING TO CHARACTER AFTER NUMBER
                                ;       R0 - COUNT OF DIGITS (0 IF END OF LINE FOUND)

103712  005000                  T2GNUM: CLR R0                          ;CLEAR DIGIT COUNT
103714  105711                          TSTB (R1)                       ;CHECK IF END OF LINE
103716  001442                          BEQ T2GNX                       ;REPORT NULL CHARACTER FOUND
103720  121127  000040                  CMPB (R1),#' '                  ;CHECK IF A SPACE
103724  001002                          BNE T2GND1                      ;IF SO, IGNORE IT
103726  005201                          INC R1
103730  000770                          BR T2GNUM
103732  005004                  T2GND1: CLR R4                          ;CLEAR NUMBER STORAGE
103734                          T2GND2: PUSH R2                         ;SAVE REGISTER
        010246                                                                  MOV R2,-(SP)
103736  112102                          MOVB (R1)+,R2                   ;GET CHARACTER
103740  162702  000060                  SUB #'0,R2                      ;CONVERT TO HEX DIGIT
103744  100431                          BMI T2GNE
103746  020227  000011                  CMP R2,#9.
103752  003410                          BLE T2GND3
103754  020227  000021                  CMP R2,#<'A-'0>
103760  103423                          BLO T2GNE
103762  020227  000026                  CMP R2,#<'F-'0>
103766  101020                          BHI T2GNE
103770  162702  000007                  SUB #<'A-'9-1>,R2
103774  006304                  T2GND3: ASL R4
103776  006304                          ASL R4
104000  006304                          ASL R4
104002  006304                          ASL R4
104004  050204                          BIS R2,R4
104006  005200                          INC R0
104010                                  POP R2
        012602                                                                  MOV (SP)+,R2
104012  105711                          TSTB (R1)
104014  001403                          BEQ T2GNX
104016  121127  000040                  CMPB (R1),#' '
104022  001344                          BNE T2GND2
104024  005700                  T2GNX:  TST R0
104026  000207                          RETURN

104030                          T2GNE:  POP R2                          ;CLEAN UP THE STACK
        012602                                                                  MOV (SP)+,R2
104032                                  POP R0
        012600                                                                  MOV (SP)+,R0
104034  000137  100632                  JMP T2CMDE
```

```
                                        ;PRINT ONE CHARACTER
                                        ;
                                        ;CALL WITH MACRO PRINT

104040  110037  065014      CPNT:   MOVB R0,ERRCHR
104044                          PUSH R1
104044  010146
                                                        MOV R1,-(SP)
104046  012701  065443          MOV #ERRONE,R1
104052  120027  000015          CMPB R0,#CR
104056  001002                  BNE 1$
104060  012701  065446          MOV #ERRNL,R1
104064  000177  160722  1$:     JMP @PTYPE
104070                  PF:     PRINTF R1,#ERRCHR
104070  012746  065014
104074  010146
104076  012746  000002
104102  010600
104104  104417
104106  062706  000006
104112  000435                  BR CPNTX
104114                  PB:     PRINTB R1,#ERRCHR
104114  012746  065014
104120  010146
104122  012746  000002
104126  010600
104130  104414
104132  062706  000006
104136  000423                  BR CPNTX
104140                  PX:     PRINTX R1,#ERRCHR
104140  012746  065014
104144  010146
104146  012746  000002
104152  010600
104154  104415
104156  062706  000006
104162  000411                  BR CPNTX
104164                  PS:     PRINTS R1,#ERRCHR
104164  012746  065014
104170  010146
104172  012746  000002
104176  010600
104200  104416
104202  062706  000006
104206  012601          CPNTX:  POP R1
104210  000207                  RETURN
```

```
                                MOV     #ERRCHR,-(SP)
                                MOV     R1,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTF
                                ADD     #6,SP

                                MOV     #ERRCHR,-(SP)
                                MOV     R1,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTB
                                ADD     #6,SP

                                MOV     #ERRCHR,-(SP)
                                MOV     R1,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTX
                                ADD     #6,SP

                                MOV     #ERRCHR,-(SP)
                                MOV     R1,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTS
                                ADD     #6,SP

                                MOV (SP)+,R1
```

```
                                        ;PRINT FORMATTED MESSAGE
                                        ;
                                        ;CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
                                        ;
104212  012737  104070  065012  LPNTF:  MOV #PF,PTYPE
104220  000413                          BR LPNT
104222  012737  104114  065012  LPNTB:  MOV #PB,PTYPE
104230  000407                          BR LPNT
104232  012737  104140  065012  LPNTX:  MOV #PX,PTYPE
104240  000403                          BR LPNT
104242  012737  104164  065012  LPNTS:  MOV #PS,PTYPE
104250                          LPNT:   PUSH <R2,R3,R4,R5>
104250  010246                                                          MOV R2,-(SP)
104252  010346                                                          MOV R3,-(SP)
104254  010446                                                          MOV R4,-(SP)
104256  010546                                                          MOV R5,-(SP)
104260  012102                          MOV (R1)+,R2            ;GET ADDRESS OF STRING
104262  010604                          MOV SP,R4               ;COMPUTE ADDRESS OF ARGUMENTS
104264  062704  000012                  ADD #10.,R4             ;  WHICH ARE NOW ON STACK (IF ANY)
104270                                  PUSH R1                 ;SAVE RETURN ADDRESS
104270  010146                                                          MOV R1,-(SP)
104272  004737  102220                  CALL OSTRNG             ;PRINT THE FORMATTED MESSAGE
104276                                  POP <R0,R5,R4,R3,R2,R1> ;RESTORE ALL REGISTERS
104276  012600                                                          MOV (SP)+,R0
104300  012605                                                          MOV (SP)+,R5
104302  012604                                                          MOV (SP)+,R4
104304  012603                                                          MOV (SP)+,R3
104306  012602                                                          MOV (SP)+,R2
104310  012601                                                          MOV (SP)+,R1
104312  062006                          ADD (R0)+,SP            ;ADJUST STACK POINTER OVER ARGUMENTS
104314  000110                          JMP @R0                 ;RETURN
```

```
                                ;PNTERR
                                ;
                                ;PRINT ERROR MESSAGE FROM DM PROGRAM REQUEST 11 OR 12.
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - MESSAGE DATA ADDRESS
                                ;       R3 - COMMAND DATA ADDRESS
                                ;OUTPUTS:
                                ;       ERROR MESSAGE PRINTED
                                ;       BIT 15 SET IN COMMAND DATA IF DRIVE HAS BEEN DROPPED

104316                          PNTERR: PUSH <R0,R1,R2>
104316  010046                                                                    MOV R0,-(SP)
104320  010146                                                                    MOV R1,-(SP)
104322  010246                                                                    MOV R2,-(SP)
104324  005764  000004                 TST 4(R4)               ;GET DRIVE NUMBER
104330  002004                          BGE 1$                  ;CHECK IF BIT 15 SET
104332  116537  000002  002074         MOVB C.UNIT(R5),L$LUN   ;IF SO, GET UNIT FROM CONTROLLER TABLE
104340  000416                          BR 2$
104342                          1$:     PUSH R4                 ;SAVE DATA ADDRESS
104342  010446                                                                    MOV R4,-(SP)
104344  016401  000004                 MOV 4(R4),R1            ;GET DRIVE NUMBER
104350  004737  103132                 CALL GTDRVT             ;GET DRIVE TABLE ADDRESS
104354  001036                          BNE     5$              ;IF UNIT DROPPED, EXIT
104356  005764  000002                 TST D.UNIT(R4)          ;SEE IF UNIT HAS BEEN DROPPED FROM TESTING
104362  100004                          BPL 3$                  ;  PROCEED IF STILL TO BE TESTED
104364  052713  100000                 BIS #BIT15,(R3)         ;TELL DM PROGRAM TO STOP TESTING THIS UNIT
104370                                  POP R4
104370  012604                                                                    MOV (SP)+,R4
104372  000423                          BR 4$
104374                          3$:     POP R4                  ;RESTORE DATA ADDRESS
104374  012604                                                                    MOV (SP)+,R4
104376  012702  065002          2$:     MOV #ERRTYP,R2          ;GET POINTER TO ERROR TABLE
104402  016412  000002                 MOV 2(R4),(R2)          ;GET ERROR TYPE
104406  006112                          ROL (R2)
104410  006112                          ROL (R2)
104412  006112                          ROL (R2)
104414  042722  177774                 BIC #^C3,(R2)+          ;CLEAR LOW 2 BITS
104420  016412  000002                 MOV 2(R4),(R2)
104424  042722  140000                 BIC #140000,(R2)+       ;MASK LOW 14 BITS
104430  005022                          CLR (R2)+               ;CLEAR MESSAGE POINTER
104432  012712  075662                 MOV #ERRRTN,(R2)        ;GET ROUTINE NUMBER
104436                                  ERROR                   ;PRINT THE ERROR MESSAGE
104436  104460                                                                     TRAP    C$ERROR
104440  000241                          CLC                     ;DRIVE HAS NOT BEEN DROPPED
104442                          4$:     POP <R2,R1,R0>
104442  012602                                                                    MOV (SP)+,R2
104444  012601                                                                    MOV (SP)+,R1
104446  012600                                                                    MOV (SP)+,R0
104450  000207                          RETURN
104452  000261                  5$:     SEC                     ;DRIVE HAS BEEN DROPPED
104454  000772                          BR      4$
```

```
                                ;LOADDM
                                ;
                                ;LOAD AND START A DM PROGRAM INTO A CONTROLLER
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;IMPLICIT INPUTS:
                                ;       DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
                                ;OUTPUTS:
                                ;       IF LOAD SUCCEEDS - Z CLEAR
                                ;                          CONTROLLER TABLE MARKED LOADED
                                ;       IF ERROR - Z SET
                                ;
104456                          LOADDM:
104456  016504  000004                MOV C.VEC(R5),R4                ;GET VECTOR OF UDA
104462                                  AND CT.VEC,R4
104462  042704  177000                                                  BIC #^C<CT.VEC>,R4
104466  010501                        MOV R5,R1                       ;GET INTERRUPT SERVICE LINK
104470  062701  000010                ADD #C.JSR,R1
104474                                  SETVEC R4,R1,#PRI07             ;SET UP INTERRUPT VECTOR
104474  012746  000340                                                  MOV     #PRI07,-(SP)
104500  010146                                                          MOV     R1,-(SP)
104502  010446                                                          MOV     R4,-(SP)
104504  012746  000003                                                  MOV     #3,-(SP)
104510  104437                                                          TRAP    C$SVEC
104512  062706  000010                                                  ADD     #10,SP
                                                                ;INITIALIZE UDA WITH SMALLEST
104516  006204                        ASR R4                          ;POSITION VECTOR FOR UDA
104520  006204                        ASR R4
104522  004737  106020                CALL UDAINT                     ; RING BUFFER AND INTERRUPTS ENABLED
104526  001566                        BEQ LOADER                      ;BRANCH IF AN ERROR
104530  004737  076060                CALL HCOMM                      ;ALLOCATE SPACE FOR HOST COMM AREA
104534  022737  000004  065050        CMP     #4,TNUM                 ; IS IT TEST 4?
104542  001402                        BEQ     3$                      ; IF SO, BRANCH
104544  004737  105110                CALL    SETDPR                  ; SET DMPROG
104550                          3$:
```

```
      2 104550  023727  065050  000001          CMP TNUM,#1                         ;IF TEST NUMBER 1
      3 104556  001440                          BEQ LOADT1                          ;  DO SPECIAL LOAD
      5 104560  017701  160250                  MOV @DMPROG,R1                      ;GET SIZE OF PROGRAM
      6 104564  012700  000002          LOADB:  MOV #OP.ESP,R0                      ;BUILD EXECUTE SUPPLIED PROGRAM COMMAND PACKET
      7 104570  004737  105302                  CALL BLDCMD
      8 104574  013764  065034  000124          MOV DMPROG,HC.CPK+P.UADR(R4)        ;LOAD MAIN PROGRAM ADDRESS
      9 104602  010164  000120                  MOV R1,HC.CPK+P.BCNT(R4)            ;  AND SIZE
     10 104606  013764  065034  000140          MOV DMPROG,HC.CPK+P.OVRL(R4)        ;LOAD OVERLAY ADDRESS
     11 104614  067764  160214  000140          ADD @DMPROG,HC.CPK+P.OVRL(R4)
     15 104622  004737  105366                  CALL SNDCMD                         ;SEND COMMAND TO UDA
     16 104626  004737  105526                  CALL WAITMS                         ;WAIT FOR MESSAGE RESPONSE
     17 104632  032764  000037  000032          BIT #ST.MSK,HC.MPK+P.STS(R4)        ;CHECK FOR ERRORS
     18 104640  001115                          BNE LOADE1
     19 104642  042765  000024  000014          BIC #CT.CMD+CT.REQ,C.FLG(R5)        ;CLEAR COMMAND OUTSTANDING FLAG
     20 104650  052765  000002  000014          BIS #CT.RN,C.FLG(R5)                ;SET DM PROGRAM RUNNING FLAG
     24 104656  000207                          RETURN
```

```
                                        ;LOAD DM PROGRAM FROM MEMORY SPACE TESTED DURING
                                        ;INITIALIZATION IN TEST 1
                                        
104660  017704  160150    LOADT1: MOV @DMPROG,R4          ;GET SIZE OF DM PROGRAM IN BYTES
104664  162704  000040            SUB #DMMAIN,R4
104670  013700  065034            MOV DMPROG,R0           ;GET ADDRESS OF DM PROGRAM
104674  062700  000040            ADD #DMMAIN,R0
104700  005001                    CLR R1                  ;START WITH OFFSET OF ZERO
                                
104702  012703  000214    LT1L1:  MOV #<HC.BSZ*2>,R3      ;GET SIZE OF BOTH BUFFERS
104706  020403                    CMP R4,R3               ;IF FEWER BYTES REMAINING IN PROGRAM
104710  103001                    BHIS LT11
104712  010403                    MOV R4,R3               ;USE ACTUAL BYTE COUNT
104714                  LT11:   PUSH R3                 ;SAVE THE BYTE COUNT
104714  010346                                                  MOV R3,-(SP)
104716  013702  065016            MOV FFREE,R2            ;GET ADDRESS OF BUFFER
104722  162702  000214            SUB #<HC.BSZ*2>,R2
104726                            PUSH R2                 ;SAVE BUFFER ADDRESS
104726  010246                                                  MOV R2,-(SP)
104730  012022          LT1L2:  MOV (R0)+,(R2)+         ;MOVE DATA TO BUFFER
104732  162703  000002            SUB #2,R3               ;COUNT BYTES
104736  001374                    BNE LT1L2
104740                            POP R2                  ;RESTORE BUFFER ADDRESS
104740  012602                                                  MOV (SP)+,R2
104742                            POP R3                  ;RESTORE BYTE COUNT
104742  012603                                                  MOV (SP)+,R3
104744  004737  104772            CALL LOAD               ;LOAD INTO UDA
104750  001455                    BEQ LOADER              ;IF ERROR, GET OUT NOW
104752  006203                    ASR R3                  ;CONVERT BYTES TO WORDS
104754  060301                    ADD R3,R1               ;INCREASE OFFSET FOR NEXT BUFFER
104756  006303                    ASL R3                  ;CONVERT WORDS TO BYTES
104760  160304                    SUB R3,R4               ;REDUCE REMAINING BYTE COUNT
104762  001347                    BNE LT1L1               ;GET NEXT BUFFER
104764  012701  000040            MOV #DMMAIN,R1          ;GET A BYTE COUNT OF HEADER ONLY
104770  000675                    BR LOADB                ;NOW START
```

```
                                        ;LOAD
                                        ;
                                        ;ISSUE DOWNLINE LOAD COMMAND TO UDA. CHECK THAT LOAD
                                        ;HAPPENS WITHOUT ERROR.
                                        ;
                                        ;INPUTS:
                                        ;       R1 - OFFSET FOR DM PROGRAM
                                        ;       R2 - ADDRESS OF BUFFER CONTAINING PROGRAM
                                        ;       R3 - SIZE OF BUFFER IN BYTES
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;OUTPUTS:
                                        ;       Z CLEAR IF NO ERROR
                                        ;       Z SET IF ERROR AND ERROR REPORTED
                                        ;
104772                                  LOAD:   PUSH <R0,R3,R4>
104772  010046                                                                  MOV R0,-(SP)
104774  010346                                                                  MOV R3,-(SP)
104776  010446                                                                  MOV R4,-(SP)
105000  012700  000031                  MOV #OP.MWR,R0          ;GET DOWNLINE LOAD COMMAND
105004  004737  105302                  CALL BLDCMD             ;BUILD COMMAND PACKET
105010  010264  000124                  MOV R2,HC.CPK+P.UADR(R4)        ;STUFF IN BUFFER ADDRESS
105014  010364  000120                  MOV R3,HC.CPK+P.BCNT(R4)        ;STUFF IN BYTE COUNT
105020  010164  000144                  MOV R1,HC.CPK+P.RGOF(R4)        ;STUFF IN OFFSET
105024  012764  000001  000140          MOV #1,HC.CPK+P.RGID(R4)        ;STUFF IN REGION ID 1
105032  004737  105366                  CALL SNDCMD             ;SEND COMMAND TO UDA
105036  004737  105526                  CALL WAITMS             ;WAIT FOR MESSAGE RESPONSE
105042  001420                          BEQ     LOADER          ; IF FAILED, EXIT
105044  032764  000037  000032          BIT #ST.MSK,HC.MPK+P.STS(R4)    ;LOOK FOR ANY ERROR
105052  001010                          BNE LOADE1
105054  042765  000004  000014          BIC #CT.CMD,C.FLG(R5)           ;CLEAR COMMAND ISSUED
105062                                  POP <R4,R3,R0>
105062  012604                                                                  MOV (SP)+,R4
105064  012603                                                                  MOV (SP)+,R3
105066  012600                                                                  MOV (SP)+,R0
105070  000244                          CLZ                     ;CLEAR Z TO INDICATE NO ERROR
105072  000207                          RETURN
```

```
                                ;UDA FAILED TO DOWNLINE LOAD DM PROGRAM
                                
105074                          LOADE1: ERRDF 34,,ERR034
105074  104455                                                  TRAP    C$ERDF
105076  000042                                                  .WORD   34
105100  000000                                                  .WORD   0
105102  075404                                                  .WORD   ERR034
105104  000264                  LOADER: SEZ                     ;SET Z TO INDICATE ERROR OCCURRED
105106  000207                          RETURN
```

```
                                        ;SETDPR
                                        ;
                                        ;SET DMPROG WITH APPROPRIATE ADDRESS
                                        ;
105110                                  SETDPR: PUSH    <R0,R4>
105110  010046                                                                    MOV R0,-(SP)
105112  010446                                                                    MOV R4,-(SP)
105114  004737  105202                         CALL    GTDUST                  ; GET DUST STATUS
105120  023764  065132  000034                 CMP     U52EXT,HC.MPK+P.DEXT(R4) ; U52?
105126  001011                                 BNE     1$                      ; IF NOT, BRANCH
105130  012737  002122  065034                 MOV     #STORAGE,DMPROG         ; SET DM52 ADDRESS
105136  032765  000040  000014                 BIT     #CT.U50,C.FLG(R5)       ; DO THEY MATCH?
105144  001413                                 BEQ     2$                      ; IF SO, EXIT
105146  000137  076350                         JMP     RNT4E                   ; ELSE REPORT ERROR
105152  013737  065036  065034  1$:            MOV     D50PRG,DMPROG           ; SET DM50 ADDRESS
105160  032765  000040  000014                 BIT     #CT.U50,C.FLG(R5)       ; DOES CONTROLLER TYPE MATCH
105166  001002                                 BNE     2$                      ; IF SO, EXIT
105170  000137  076350                         JMP     RNT4E                   ; ELSE, REPORT ERROR
105174                                  2$:     POP     <R4,R0>                 ; RESET
105174  012604                                                                    MOV (SP)+,R4
105176  012600                                                                    MOV (SP)+,R0
105200  000207                                 RETURN
```

```
                                        ;GTDUST
                                        ;
                                        ;GET DUST STATUS
                                        ;
 5 105202  016504  000004       GTDUST: MOV C.VEC(R5),R4                ;GET VECTOR OF UDA
 6 105206                               AND CT.VEC,R4
   105206  042704  177000                                                     BIC #^C<CT.VEC>,R4
 7 105212  010501                       MOV R5,R1                       ;GET INTERRUPT SERVICE LINK
 8 105214  062701  000010               ADD #C.JSR,R1
 9 105220                               SETVEC R4,R1,#PRI07             ;SET UP INTERRUPT VECTOR
   105220  012746  000340                                                     MOV     #PRI07,-(SP)
   105224  010146                                                            MOV     R1,-(SP)
   105226  010446                                                            MOV     R4,-(SP)
   105230  012746  000003                                                     MOV     #3,-(SP)
   105234  104437                                                            TRAP    C$SVEC
   105236  062706  000010                                                     ADD     #10,SP
10                                                                      ;INITIALIZE UDA WITH SMALLEST
11 105242  006204                       ASR R4                          ;POSITION VECTOR FOR UDA
12 105244  006204                       ASR R4
13 105246  004737  106020               CALL UDAINT                     ; RING BUFFER AND INTERRUPTS ENABLED
14 105252  001714                       BEQ LOADER                      ;BRANCH IF AN ERROR
15 105254  004737  076060               CALL HCOMM                      ;ALLOCATE SPACE FOR HOST COMM AREA
16 105260  012700  000001               MOV     #OP.GSS,R0              ; R0 HAS OPCODE
17 105264  004737  105302               CALL    BLDCMD
18 105270  004737  105366               CALL    SNDCMD
19 105274  004737  105526               CALL    WAITMS
20 105300  000207                       RETURN
```

```
                                ;BLDCMD
                                ;
                                ;BUILD A COMMAND IN COMMAND PACKET
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R0 - COMMAND CODE
                                ;OUTPUTS:
                                ;       R4 - ADDRESS OF HOST COMM AREA
                                ;       COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL OTHER FIELDS CLEARED.
                                ;       CMD REFERRENCE NUMBER IN CONTROLLER TABLE INCREMENTED AND RESULT
                                ;       IN COMMAND PACKET.
                                ;       R0 - CONTENTS DESTROYED
                                ;
105302                          BLDCMD: PUSH <R1,R0>
105302  010146                                                                  MOV R1,-(SP)
105304  010046                                                                  MOV R0,-(SP)
105306  016504  000016                 MOV C.RING(R5),R4       ;GET ADDRESS OF HOST COMM AREA
105312  010400                         MOV R4,R0               ;COPY TO R0
105314  062700  000100                 ADD #HC.CEV,R0          ;COMPUTE ADDRESS OF COMMAND ENVELOPE
105320  012720  000060                 MOV #HC.PSZ,(R0)+       ;LOAD PACKET LENGTH
105324  012701  001000                 MOV #DUP,R1             ;LOAD DIAG CIRCUIT IDENTIFIER
105330  022716  000031                 CMP #OP.MWR,(SP)        ;IF CODE IS MAINTENANCE WRITE
105334  001002                         BNE BLDC0               ;  GET OTHER CIRCUIT IDENTIFIER
105336  012701  177777                 MOV #DIAG,R1
105342  010120                 BLDC0:  MOV R1,(R0)+            ;PUT IDENTIFIER INTO PACKET
105344  012701  000030                 MOV #<HC.PSZ>/2,R1      ;GET WORDS TO CLEAR
105350  005020                 BLDC1:  CLR (R0)+               ;CLEAR PACKET
105352  005301                         DEC R1
105354  001375                         BNE BLDC1
105356                                 POP HC.CPK+P.OPCD(R4)   ;PUT OPCODE IN PACKET
105356  012664  000114                                                          MOV (SP)+,HC.CPK+P.OPCD(R4)
105362                                 POP R1                  ;RESTORE R1
105362  012601                                                                  MOV (SP)+,R1
105364  000207                         RETURN
```

```
                                        ;SNDCMD
                                        ;
                                        ;SEND A COMMAND TO THE UDA.
                                        ;CLEAR THE RESPONSE PACKET. MARK BOTH PACKETS AVAILABLE TO THE
                                        ;UDA. SET COMMAND ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE
                                        ;TIMEOUT COUNTER.
                                        ;
                                        ;INPUTS:
                                        ;       R5 - CONTROLLER TABLE ADDRESS
                                        ;OUTPUTS:
                                        ;       R4 - ADDRESS OF HOST COMM AREA


105366                                  SNDCMD: PUSH <R0,R1>
105366  010046                                                                    MOV R0,-(SP)
105370  010146                                                                    MOV R1,-(SP)
105372  016504  000016                        MOV C.RING(R5),R4                ;LOAD R4 WITH HOST COMM AREA ADDRESS
105376  005265  000044                        INC C.REF(R5)                    ;INCREMENT CMD REFERENCE NUMBER
105402  016564  000044  000104                MOV C.REF(R5),HC.CPK+P.CRF(R4)   ;PUT IN PACKET
105410  012700  000014                        MOV #HC.MEV,R0                   ;POINT TO MESSAGE ENVELOPE
105414  060400                                ADD R4,R0
105416  012701  000032                        MOV #<HC.PSZ+HC.ESZ>/2,R1        ;SIZE OF MESSAGE PACKET
105422  005020                          SNDC1:  CLR (R0)+                        ;CLEAR ENTIRE MESSAGE PACKET
105424  005301                                  DEC R1
105426  001375                                  BNE SNDC1
105430  012764  140000  000006                MOV #RG.OWN+RG.FLG,HC.MCT(R4)    ;MARK MESSAGE PACKET AVAILABLE
105436  012764  100000  000012                MOV #RG.OWN,HC.CCT(R4)           ;MARK COMMAND TO UDA
105444  005775  000000                        TST @(R5)                        ;TELL UDA COMMAND IS THERE
105450  052765  000004  000014                BIS #CT.CMD,C.FLG(R5)            ;MARK COMMAND ISSUED
105456                                  POP     <R1,R0>
105456  012601                                                                  MOV (SP)+,R1
105460  012600                                                                  MOV (SP)+,R0
105462  000207                          RETURN
```

```
                                ;CLRBUF
                                ;
                                ;CLEAR THE SPECIFIED DATA BUFFER IN THE HOST COMM AREA
                                ;AND LOAD BUFFER DESCRIPTOR IN COMMAND PACKET TO THE BUFFER
                                ;
                                ;INPUTS:
                                ;       R5 - CONTROLLER TABLE ADDRESS
                                ;       R4 - ADDRESS OF HOST COMM AREA
                                ;       R0 - OFFSET INTO HOST COMM AREA TO DATA BUFFER
                                ;OUTPUTS:
                                ;       DATA BUFFER CLEARED
                                ;       COMMAND PACKET POINTING TO BUFFER
                                ;       BYTE COUNT SET TO SIZE OF BUFFER
                                ;       R4 - ADDRESS OF DATA BUFFER
                                ;
105464                          CLRBUF: PUSH <R0,R1>
105464  010046                                                                    MOV R0,-(SP)
105466  010146                                                                    MOV R1,-(SP)
105470  060400                          ADD R4,R0                       ;ADD START OF HOST COMM AREA TO OFFSET
105472  010064  000124                 MOV R0,HC.CPK+P.UADR(R4)        ;PUT BUFFER ADDRESS IN COMMAND PACKET
105476  012764  000106  000120         MOV #HC.BSZ,HC.CPK+P.BCNT(R4)   ;PUT SIZE OF BUFFER IN COMMAND PACKET
105504  010004                          MOV R0,R4                       ;PUT BUFFER ADDRESS IN R4
105506  012701  000043                 MOV #HC.BSZ/2,R1                ;GET SIZE OF BUFFER IN WORDS
105512  005020                  CLRBFL: CLR (R0)+                       ;CLEAR ALL THE WORDS
105514  005301                          DEC R1
105516  001375                          BNE CLRBFL
105520                                  POP <R1,R0>
105520  012601                                                                    MOV (SP)+,R1
105522  012600                                                                    MOV (SP)+,R0
105524  000207                          RETURN
```

```
                                        ;WAITMS
                                        ;
                                        ;WAIT FOR UDA TO RESPOND WITH A MESSAGE PACKET
                                        ;
                                        ;INPUTS:
                                        ;       R5 - ADDRESS OF CONTROLLER TABLE
                                        ;OUTPUTS:
                                        ;       Z CLEAR IF NO ERROR
                                        ;       Z SET IF ERROR, MESSAGE PRINTED
                                        ;
105526                                  WAITMS: PUSH <R0,R1>
105526  010046                                                                    MOV R0,-(SP)
105530  010146                                                                    MOV R1,-(SP)
105532  012700  000036                       MOV #30.,R0             ;SET TIME OUT VALUE OF 30 SECONDS
105536  010501                               MOV R5,R1               ;POINT TO TIME OUT COUNTER
105540  062701  000040                       ADD #C.TO,R1
105544  004737  105736                       CALL SETTO
105550  011500                          1$:  MOV (R5),R0             ;GET ADDRESS OF UDAIP REGISTER
105552  032765  000010  000014               BIT #CT.MSG,C.FLG(R5)   ;LOOK IF INTERRUPT OCCURRED
105560  001030                               BNE 3$                  ;BRANCH IF SO
105562  016001  000002                       MOV 2(R0),R1            ;LOOK AT UDASA REGISTER
105566  001034                               BNE 4$                  ;BRANCH IF ERROR CODE PRESENT
105570                                       BREAK
105570  104422                                                                  TRAP    C$BRK
105572  005737  065222                       TST KW.CSR              ;SEE IF A CLOCK ON SYSTEM
105576  001764                               BEQ 1$
105600  023765  065234  000042               CMP KW.EL+2,C.TOH(R5)   ;CHECK IF TIMEOUT HAS HAPPENED
105606  101005                               BHI 2$
105610  001357                               BNE 1$
105612  023765  065232  000040               CMP KW.EL,C.TO(R5)
105620  103753                               BLO 1$
105622                                  2$:  ERRDF 36,,ERR036
105622  104455                                                                  TRAP    C$ERDF
105624  000044                                                                  .WORD   36
105626  000000                                                                  .WORD   0
105630  075430                                                                  .WORD   ERR036
105632                                       POP <R1,R0>
105632  012601                                                                  MOV (SP)+,R1
105634  012600                                                                  MOV (SP)+,R0
105636  000264                               SEZ
105640  000207                               RETURN
```

```
      1 105642  042765  000010  000014  3$:     BIC #CT.MSG,C.FLG(R5)          ;CLEAR MESSAGE RECEIVED FLAG
      2 105650                                  POP <R1,R0>
        105650  012601                                                          MOV (SP)+,R1
        105652  012600                                                          MOV (SP)+,R0
      3 105654  000244                          CLZ                           ;GIVE NO ERROR RETURN
      4 105656  000207                          RETURN
      5 105660                            4$:   ERRDF 37,,ERR037
        105660  104455                                                          TRAP    C$ERDF
        105662  000045                                                          .WORD   37
        105664  000000                                                          .WORD   0
        105666  075442                                                          .WORD   ERR037
      6 105670                                  POP <R1,R0>
        105670  012601                                                          MOV (SP)+,R1
        105672  012600                                                          MOV (SP)+,R0
      7 105674  000264                          SEZ
      8 105676  000207                          RETURN
```

```
                                        ;APRINT
                                        ;
                                        ;CONVERT AN 18 BIT ADDRESS STORED IN TWO WORDS INTO A FORMAT
                                        ;THAT WILL ALLOW PRINTING OF THE 18 BIT NUMBER.
                                        ;
                                        ;INPUTS:
                                        ;       R0 - ADDRESS OF TWO WORD BLOCK CONTAINING ADDRESS.
                                        ;                FIRST WORD CONTAINING LOW 16 BITS.
                                        ;                SECOND WORD CONTAINING HIGH 2 BITS.
                                        ;OUTPUTS:
                                        ;       R1 - HIGH 3 BITS OF ADDRESS
                                        ;       R2 - LOW 15 BITS OF ADDRESS
                                        ;
105700  016001  000002        APRINT: MOV 2(R0),R1                    ;GET HIGH 2 BITS
105704  006301                        ASL R1                          ;SHIFT LEFT
105706  011002                        MOV (R0),R2                     ;GET LOW 16 BITS
105710  100001                        BPL APRIZ                       ;IF 16TH BIT SET
105712  005201                        INC R1                          ;PLACE IT IN WITH HIGH 2 BITS
105714  000207                APRIZ:  RETURN
```

```
                                ;NXMI
                                ;
                                ;NON-EXISTANT MEMORY SERVICE ROUTINE
                                ;
                                ;INPUTS:
                                ;       NXMAD SET TO ZERO
                                ;OUTPUTS:
                                ;       NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
                                
105716                          BGNSRV NXMI
105716                                                                  NXMI::
                                
105716  012737  177777  065242        MOV #-1,NXMAD
                                
105724                          ENDSRV
105724                                                                  L10040:
105724  000002                                                                  RTI
```

```
                        ;UDASRV
                        ;
                        ;UDA INTERRUPT SERVICE ROUTINE. MARKS UDA CONTROLLER TABLE THAT AN
                        ;INTERRUPT HAS BEEN RECEIVED.
                        ;
                        ;THIS ROUTINE IS CALLED BY A [JSR R0,UDASRV] INSTRUCTION FROM WITHIN
                        ;THE CONTROLLER TABLE. THE PC STORED IN R0 IS THE ADDRESS OF THE C.FLG
                        ;WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
                        ;OF R0 FOLLOWED BY THE INTERRUPTED PC AND PS.
                        ;
                        ;INPUTS:
                        ;       R0 - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
                        ;       STACK - SAVED CONTENTS OF R0
                        ;OUTPUTS:
                        ;       CT.CMD CLEARED AND CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
                        ;       R0 - RESTORED FROM STACK

105726                  BGNSRV UDASRV
105726                                                          UDASRV::
105726  052710  000010          BIS #CT.MSG,(R0)                ;SET CT.MSG
105732                          POP R0                          ;RESTORE R0
105732  012600                                                          MOV (SP)+,R0
105734                  ENDSRV
105734                                                          L10041:
105734  000002                                                          RTI
```

```
                                ;SETTO
                                ;
                                ;SET TIMEOUT COUNTER TO SOME NUMBER OF SECONDS FROM CURRENT TIME.
                                ;
                                ;INPUTS:
                                ;       R0 - NUMBER OF SECONDS FOR TIMEOUT
                                ;       R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT
                                ;OUTPUTS:
                                ;       R0 - CONTENTS DESTROYED
                                ;       R1 - INCREMENTED BY 2

                                ;COMPUTE CLOCK TICKS TIL TIMEOUT

 27 105736                      SETTO:  PUSH <R2,R3>
    105736  010246                                                      MOV R2,-(SP)
    105740  010346                                                      MOV R3,-(SP)
 29 105742  005002                      CLR R2                         ;CLEAR PRODUCT
 30 105744  013703  065230              MOV KW.HZ,R3                   ;GET MULTIPLICAND
 45 105750  006200              SETOO:  ASR R0                         ;SHIFT MULTIPLIER TO RIGHT
 46 105752  103001                      BCC SETO1                      ;IF A ONE BIT SHIFTED OUT
 47 105754  060302                      ADD R3,R2                      ;  ADD MULTIPLICAND TO PRODUCT
 48 105756  006303              SETO1:  ASL R3                         ;DOUBLE THE MULTIPLICAND
 49 105760  005700                      TST R0
 50 105762  001372                      BNE SETOO                      ;CONTINUE UNTIL MULTIPLIER IS ZERO
 52                             ;GET CURRENT TIME
 53
 54 105764  013700  065232      SETO2:  MOV KW.EL,R0                   ;GET TIME
 55 105770  013703  065234              MOV KW.EL+2,R3
 56 105774  020037  065232              CMP R0,KW.EL                   ;IF CHANGED DURING RETRIEVAL
 57 106000  001371                      BNE SETO2                      ;  GET IT AGAIN
 58
 59                             ;ADD TIME TIL TIMEOUT
 60
 61 106002  060200                      ADD R2,R0                      ;ADD
 62 106004  005503                      ADC R3
 66
 67                             ;PUT RESULT IN STORAGE
 68
 69 106006  010021                      MOV R0,(R1)+
 70 106010  010311                      MOV R3,(R1)
 71
 75 106012                              POP <R3,R2>
    106012  012603                                                      MOV (SP)+,R3
    106014  012602                                                      MOV (SP)+,R2
 77 106016  000207                      RETURN
```

```
                                        ;UDAINT
                                        ;
                                        ;FUNCTIONAL DESCRIPTION:
                                        ;       SUBROUTINE TO INITIALIZE A UDA AND BRING IT ON-LINE.
                                        ;       ALL STEPS ARE CHECKED. AN ERROR MESSAGE IS REPORTED IF ANY ERROR
                                        ;       DETECTED.
                                        ;
                                        ;INPUTS:
                                        ;       R5 - ADDRESS OF CONTROLLER TABLE.
                                        ;       R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO UDA
                                        ;IMPLICIT INPUTS:
                                        ;       FFREE - FIRST FREE ADDRESS OF MEMORY. THIS ADDRESS IS GIVEN TO UDA
                                        ;               AS START OF RING BUFFER.
                                        ;       FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS.
                                        ;OUTPUTS:
                                        ;       CONDITION Z - SET IF ANY ERROR REPORTED. CLEAR IF NO ERROR.
                                        ;       R1 - SIZE OF RING BUFFER IN WORDS IF NO ERROR.
                                        ;       R4 - ADDRESS OF UDAIP REGISTER IN UDA
                                        ;       R5 - UNCHANGED.

                                        ;CHECK IF ENOUGH FREE MEMORY FOR RING BUFFER

106020  010400                  UDAINT: MOV R4,R0                       ;GET MESSAGE LENGTH
106022  000300                          SWAB R0
106024  042700  177770                  BIC #177770,R0
106030  004737  106764                  CALL CLOG                       ;COMPUTE LOGARITHMIC VALUE
106034  010102                          MOV R1,R2                       ;SAVE RESULT IN R2
106036  010400                          MOV R4,R0                       ;GET COMMAND LENGTH
106040  000300                          SWAB R0
106042  006000                          ROR R0
106044  006000                          ROR R0
106046  006000                          ROR R0
106050  042700  177770                  BIC #177770,R0
106054  004737  106764                  CALL CLOG                       ;COMPUTE LOGARITHMIC VALUE
106060  060201                          ADD R2,R1                       ;ADD THE TWO RESULTS
106062  006301                          ASL R1                          ;MULTIPLY BY 2 WORDS PER RING
106064  062701  000002                  ADD #HC.ISZ/2,R1                ;ADD SPACE FOR INTERRUPT INDICATORS
106070  020137  065020                  CMP R1,FSIZE                    ;COMPARE WITH SIZE OF FREE MEMORY
106074  101402                          BLOS UDAI1
106076  000137  076022                  JMP FMERR                       ;FATAL ERROR IF NOT ENOUGH MEMORY
```

```
                                ;FILL HOST COMMUNICATION AREA WITH ALL ONES
                                ;
106102  013702  065016  UDAI1:  MOV FFREE,R2          ;GET FIRST ADDRESS OF RING BUFFER
106106  010103                 MOV R1,R3             ;GET SIZE OF RING BUFFER
106110  012722  177777  UDAI1L: MOV #-1,(R2)+         ;WRITE ONES TO BUFFER
106114  005303                 DEC R3                ;COUNT THE WORDS IN BUFFER
106116  003374                 BGT UDAI1L            ;LOOP UNTIL ENTIRE BUFFER WRITTEN
                                ;
                                ;DO THE INITIALIZATION
                                ;
106120  004737  106270         CALL UDAIST           ;DO FIRST THREE STEPS
106124  103457                 BCS UDAIEX            ;GET OUT IF UDA MICROCODE REPORTED FAILURE
106126  012364  000002         MOV (R3)+,2(R4)       ;WRITE NEXT WORD TO UDASA REGISTER
106132  012700  000310         MOV #200.,R0          ;GET TRY COUNTER
106136  016402  000002  UDAI1A: MOV 2(R4),R2         ;LOOK AT UDASA
106142  001410                 BEQ UDAI1C
106144  100005                 BPL UDAI1B
106146                         ERRDF 24,,ERR024
106146  104455                                                  TRAP    C$ERDF
106150  000030                                                  .WORD   24
106152  000000                                                  .WORD   0
106154  075214                                                  .WORD   ERR024
106156  000442                 BR UDAIEX
106160  005300         UDAI1B: DEC R0
106162  001365                 BNE UDAI1A
106164  010264  000002  UDAI1C: MOV R2,2(R4)         ;WRITE 0 TO UDASA (PURGE)
106170  011402                 MOV (R4),R2           ;READ FROM UDAIP (POLL)
106172  004737  106624         CALL UDARSP           ;WAIT FOR STEP OR ERROR BIT
106176  103432                 BCS UDAIEX            ;GET OUT IF UDA MICROCODE REPORTED FAILURE
106200                         PUSH    R1
106200  010146                                                  MOV R1,-(SP)
106202  004733                 CALL    @(R3)+        ; CALL LAST ROUTINE
106204                         POP     R1
106204  012601                                                  MOV (SP)+,R1
                                ;
                                ;CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
                                ;
106206  013702  065016  UDAI2:  MOV FFREE,R2          ;GET FIRST ADDRESS OF RING BUFFER
106212  010103                 MOV R1,R3             ;GET SIZE OF RING BUFFER
106214  005722         UDAI2L: TST (R2)+             ;CHECK WORD IN BUFFER
106216  001003                 BNE UDAI2E            ;GO TO ERROR REPORTER IF NOT ZERO
106220  005303                 DEC R3                ;COUNT THE WORDS IN BUFFER
106222  003374                 BGT UDAI2L            ;LOOP UNTIL ALL WORDS CHECKED
106224  000405                 BR UDAI3
                                ;
106226                 UDAI2E: ERRDF 23,,ERR023      ;REPORT BUFFER NOT CLEARED
106226  104455                                                  TRAP    C$ERDF
106230  000027                                                  .WORD   23
106232  000000                                                  .WORD   0
106234  075130                                                  .WORD   ERR023
106236  000412                 BR UDAIEX
```

```
                                        ;SEND GO BIT TO UDASA REGISTER TO END INITIALIZATION
                                        
003240                                  UDAI3:
012500  000006                                  MOV C.BST(R5),R0                ;GET BURST VALUE
006300                                          ASL R0                          ;SHIFT TO POSITION
006300                                          ASL R0
052700  000001                                  BIS #SA.GO,R0                   ;SET THE GO BIT
010064  000002                                  MOV R0,2(R4)                    ;SEND TO UDA
000244                                          CLZ                             ;CLEAR Z AS NO ERROR INDICATION
000207                                          RETURN
                                        
                                        ;ERROR RETURN
                                        
000264                                  UDAIEX: SEZ                             ;SET Z TO INDICATE ERROR OCCURRED
000207                                          RETURN
```

```
                                ;UDAIST
                                ;
                                ;START THE INITIALIZATION PROCESS ON THE SELECTED UDA.
                                ;STOP BEFORE WRITING THE THIRD WORD SO UDA DOES NOT
                                ;ATTEMPT ANY UNIBUS TRANSFERS.
                                ;
                                ;INPUTS:
                                ;       R5 - ADDRESS OF CONTROLLER TABLE
                                ;       R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO UDA

                                ;LOAD TABLE OF DATA TO SEND TO UDASA REGISTER

 106270                         UDAIST: BREAK
 106270  104422                                                                   TRAP    C$BRK
 106272                                 PUSH R1
 106272  010146                                                                   MOV R1,-(SP)
 106274  052704  100000                 BIS #SA.STP,R4          ;SET STEP BIT IN DATA WORD
 106300  010437  106472                 MOV R4,UDAID1           ;LOAD LENGTH AND INTERRUPT VECTOR
 106304  013737  065016  106476         MOV FFREE,UDAID2        ;LOAD MEMORY ADDRESS
 106312  062737  000004  106476         ADD #HC.MSG,UDAID2      ;  OF FIRST RESPONSE RING

                                ;START THE INITIALIZATION BY WRITING TO UDAIP REGISTER

 106320  016504  000000                 MOV C.UADR(R5),R4       ;GET ADDRESS OF UDAIP REGISTER
 106324  005037  065242                 CLR NXMAD               ;CLEAR MEMORY ERROR FLAG
 106330                                 SETVEC #4,#NXMI,#PRI07  ;SET UP VECTOR 4
 106330  012746  000340                                                           MOV     #PRI07,-(SP)
 106334  012746  105716                                                           MOV     #NXMI,-(SP)
 106340  012746  000004                                                           MOV     #4,-(SP)
 106344  012746  000003                                                           MOV     #3,-(SP)
 106350  104437                                                                   TRAP    C$SVEC
 106352  062706  000010                                                           ADD     #10,SP
 106356  005764  000002                 TST 2(R4)               ;ACCESS UDASA REGISTER
 106362  005014                         CLR (R4)                ;WRITE TO UDAIP
 106364                                 CLRVEC #4               ;GIVE UP THE VECTOR
 106364  012700  000004                                                           MOV     #4,R0
 106370  104436                                                                   TRAP    C$CVEC
 106372  005737  065242                 TST NXMAD               ;SEE IF A MEMORY ERROR OCCURRED
 106376  001406                         BEQ UDAISG
 106400                                 ERRDF 20,,ERR020
 106400  104455                                                                   TRAP    C$ERDF
 106402  000024                                                                   .WORD   20
 106404  000000                                                                   .WORD   0
 106406  075006                                                                   .WORD   ERR020
 106410  000261                         SEC
 106412  000424                         BR UDAISE
```

```
                                        ;SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION

106414  012737  004000  106762  UDAISG: MOV #SA.S1,UDARSD               ;STORE RESPONSE MASK
106422  012703  106470                  MOV #UDAIDT,R3                  ;AND INDEX TO TABLE

                                        ;WAIT FOR AND CHECK RESPONSE DATA

106426  004737  106624          UDAISL: CALL UDARSP                     ;WAIT FOR STEP OR ERROR BITS
106432  103414                          BCS UDAISE                      ;EXIT IF ERROR
106434  004733                          CALL @(R3)+                     ;CALL RESPONSE CHECKER FOR STEP
106436  103412                          BCS UDAISE                      ;GET OUT IF ERROR
106440  006337  106762                  ASL UDARSD                      ;SHIFT TO NEXT STEP BIT
106444  032737  040000  106762          BIT #SA.S4,UDARSD               ;CHECK IF NOW AT STEP 4
106452  001003                          BNE UDAISX                      ;GET OUT IF SO
106454  012364  000002                  MOV (R3)+,2(R4)                 ;WRITE DATA TO UDASA REGISTER
106460  000762                          BR UDAISL                       ;STAY IN LOOP

106462  000241          UDAISX: CLC                                     ;CLEAR CARRY FOR NO ERROR INDICATION
106464                  UDAISE: POP R1
106464  012601                                                          MOV (SP)+,R1
106466  000207                  RETURN
```

```
                                ;DATA TO BE SENT AND RECEIVED BY UDA INITIALIZATION

106470  106506                  UDAIDT: .WORD UDAIR1                    ;FIRST WORD RESPONSE CHECK ROUTINE
106472  000000                  UDAID1: .WORD 0                         ;FIRST WORD TO SEND TO UDASA
106474  106514                          .WORD UDAIR2                    ;SECOND WORD RESPONSE CHECK ROUTINE
106476  000000                  UDAID2: .WORD 0                         ;SECOND WORD TO SEND TO UDASA
106500  106534                          .WORD UDAIR3                    ;THIRD WORD RESPONSE CHECK ROUTINE
106502  100000                  UDAID3: .WORD SA.TST                    ;THIRD WORD TO SEND TO UDASA
106504  106552                          .WORD UDAIR4                    ;FOURTH WORD RESPONSE CHECK ROUTINE

                                ;RESPONSE CHECK FOR FIRST WORD FROM UDASA
                                ;CHECK FOR PROPER CONTROLLER TYPE

106506  012701  004400          UDAIR1: MOV #SA.S1+SA.DI,R1             ;SET STEP ONE BIT
106512  000434                          BR UDAIRC                       ;NOW COMPARE

                                ;RESPONSE CHECK FOR SECOND WORD FROM UDASA
                                ;CHECK FOR ECHO OF INTI AND VECTOR

106514  013701  106472          UDAIR2: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
106520  000301                          SWAB R1                         ;GET HIGH 8 BITS
106522  042701  177400                  BIC #177400,R1
106526  052701  010000                  BIS #SA.S2,R1                   ;SET STEP 2 BIT
106532  000424                          BR UDAIRC                       ;NOW COMPARE

                                ;RESPONSE CHECK FOR THIRD WORD FROM UDASA
                                ;CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS

106534  013701  106472          UDAIR3: MOV UDAID1,R1                   ;GET WORD SENT TO UDASA
106540  042701  177400                  BIC #177400,R1                  ;JUST LOW 8 BITS
106544  052701  020000                  BIS #SA.S3,R1                   ;SET STEP 3 BIT
106550  000415                          BR UDAIRC                       ;NOW COMPARE

                                ;RESPONSE CHECK FOR FOURTH WORD FROM UDASA
                                ;CHECK FOR ECHO OF PURGE AND LFAIL BITS

106552  010201                  UDAIR4: MOV R2,R1                       ;GET RESPONSE FROM UDA
106554  042701  137400                  BIC #^C<SA.S4+SA.CNT+SA.MCV>,R1 ;KEEP MICROCODE VERSION AND STEP 4
106560  032701  000360                  BIT     #SA.CNT,R1              ; CHECK WITH CONTROLLER MODEL
106564  001404                          BEQ     1$                      ; IF ZERO, UDA50(M7161)
106566  042765  000040  000014          BIC     #CT.U50,C.FLG(R5)       ; ELSE, UDA52(M7485)
106574  000403                          BR      2$                      ; AND BRANCH
106576  052765  000040  000014  1$:     BIS     #CT.U50,C.FLG(R5)
106604                          2$:

                                ;COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2

106604  020102                  UDAIRC: CMP R1,R2                       ;COMPARE THE DATA
106606  001405                          BEQ UDAIRX                      ;EXIT IF COMPARED CORRECTLY
106610                                  ERRDF 25,,ERR025                ;REPORT ERROR
106610  104455                                                                  TRAP    C$ERDF
106612  000031                                                                  .WORD   25
106614  000000                                                                  .WORD   0
106616  075230                                                                  .WORD   ERR025
106620  000261                          SEC
106622  000207                  UDAIRX: RETURN
```

```
                                        ;UDARSP
                                        ;
                                        ;WAIT FOR UDA TO RESPOND WITH DATA IN UDASA REGISTER.
                                        ;EITHER STEP BIT FROM MASK IN LOCATION UDARSD OR ERROR BIT
                                        ;WILL CAUSE A TERMINATION.
                                        ;AN ERROR MESSAGE WILL BE PRINTED IF THE UDA DOES NOT RESPOND
                                        ;IN 10 SECONDS OR IF ERROR SETS.
                                        ;
                                        ;INPUTS:
                                        ;       UDASRD - MASK OF STEP BIT TO LOOK FOR
                                        ;       R5 - ADDRESS OF CONTROLLER TABLE
                                        ;       R4 - ADDRESS OF UDAIP REGISTER
                                        ;OUTPUTS:
                                        ;       ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
                                        ;       R2 - DATA FROM UDASA REGISTER
                                        ;       CARRY SET IF ERROR BIT SETS OR TIME OUT
                                        ;
106624                          UDARSP: PUSH R1
106624  010146                                                          MOV R1,-(SP)
106626  052737  100000  106762          BIS #SA.ERR,UDARSD              ;SET ERROR BIT IN MASK WORD
106634  012700  000012                  MOV #10.,R0                     ;SET UP FOR 10 SECOND TIMEOUT
106640  010501                          MOV R5,R1                       ;POINT TO COUNTER IN CONTROLLER TABLE
106642  062701  000040                  ADD #C.TO,R1
106646  004737  105736                  CALL SETTO
106652                                  POP R1
106652  012601                                                          MOV (SP)+,R1
106654  033764  106762  000002  UDARS1: BIT UDARSD,2(R4)                ;LOOK AT ERROR AND STEP BIT
106662  001024                          BNE UDARS2                      ;BRANCH IF EITHER SET
106664                                  BREAK
106664  104422                                                          TRAP    C$BRK
106666  005737  065222                  TST KW.CSR                      ;SEE IF CLOCK ON SYSTEM
106672  001770                          BEQ UDARS1
106674  023765  065234  000042          CMP KW.EL+2,C.TOH(R5)           ;CHECK IF TIME OUT OCCURRED
106702  101005                          BHI 1$
106704  001363                          BNE UDARS1
106706  023765  065232  000040          CMP KW.EL,C.TO(R5)
106714  103757                          BLO UDARS1
106716  016402  000002          1$:     MOV 2(R4),R2                    ;GET REGISTER CONTENTS
106722                                  ERRDF 22,,ERR022                ;REPORT TIME OUT ERROR
106722  104455                                                          TRAP    C$ERDF
106724  000026                                                          .WORD   22
106726  000000                                                          .WORD   0
106730  075062                                                          .WORD   ERR022
106732  000407                          BR UDARSE
```

```
                                        ;CHECK IF ERROR BIT SET
                                        
106734  016402  000002    UDARS2: MOV 2(R4),R2            ;GET REGISTER CONTENTS
106740  100006                    BPL UDARSX              ;EXIT IF ERROR NOT SET
106742                            ERRDF 21,,ERR021        ;REPORT ERROR INFO
106742  104455                                                    TRAP    C$ERDF
106744  000025                                                    .WORD   21
106746  000000                                                    .WORD   0
106750  075020                                                    .WORD   ERR021
106752  000261    UDARSE: SEC
106754  000207            RETURN
                                        
                                        ;NORMAL EXIT
                                        
106756  000241    UDARSX: CLC                             ;CLEAR CARRY AS NO ERROR INDICATION
106760  000207            RETURN
                                        
                                        ;LOCATION FOR STEP BIT MASK
                                        
106762  000000    UDARSD: .WORD 0                         ;LOAD BY CALLING ROUTINE
```

```
                                ;CLOG
                                ;
                                ;COMPUTE LOGARITHMIC VALUE OF NUMBER TO BASE 2.
                                ;
                                ;INPUTS:
                                ;       R0 - LOGARITHM TO BE CONVERTED
                                ;OUTPUTS:
                                ;       R1 - VALUE OF 2 RAISED TO POWER OF INPUT NUMBER
                                
106764                          CLOG:   PUSH R0
106764  010046                                                            MOV R0,-(SP)
106766  005001                          CLR R1                  ;SET UP ZERO START VALUE
106770  000261                          SEC                     ;WITH CARRY READY TO SHIFT IN
106772  006101                  CLOGLP: ROL R1                  ;SHIFT TO LEFT
106774  005300                          DEC R0                  ; UNTIL R0
106776  100375                          BPL CLOGLP              ; GOES NEGATIVE
107000                                  POP R0
107000  012600                                                            MOV (SP)+,R0
107002  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;RDDLL
                                        ;
                                        ;READ DISK DRIVE DOWNLINE LOAD PROGRAM INTO MEMORY
                                        ;
                                        ;INPUTS:
                                        ;       DLLNAM - NAME OF PROGRAM IN RAD50 (TWO WORDS)
                                        ;OUTPUTS:
                                        ;       FREE MEMORY CONTAINING PROGRAM
                                        ;       CARRY CLEAR IF NO ERROR, CARRY SET IF PROGRAM NOT FOUND

107004  012701  000012          RDDLL:  MOV #10.,R1                     ;TYPE OF PROGRAM IN DATA FILE
107010  004737  107044                  CALL RDREC                      ;READ PROGRAM INTO MEMORY
107014  006101                          ROL R1                          ;PRESERVE CARRY STATE IN R1
107016  004737  107026                  CALL CLOSEF                     ;  WHILE CLOSING THE DATA FILE
107022  006001                          ROR R1                          ;  AS NORMAL POSITION IS LOST
107024  000207                          RETURN
```

```
                                ;CLOSEF
                                ;
                                ;CLOSE DATA FILE FOR DM PROGRAMS
                                ;
                                ;INPUTS:
                                ;       FILOPN - ZERO IF FILE NOT OPEN
                                ;OUTPUTS:
                                ;       NONE
                                ;
107026  005737  065100          CLOSEF: TST FILOPN                      ;SEE IF FILE CURRENTLY OPEN
107032  001403                          BEQ 1$
107034                                  CLOSE                           ; IF SO, CLOSE IT
107034  104435                                                                  TRAP    C$CLOS
107036  005037  065100                  CLR FILOPN                      ;AND MARK AS SO
107042  000207                  1$:     RETURN
```

```
                                ;RDREC
                                ;
                                ;READ A RECORD FROM THE INPUT FILE. PLACE DATA INTO FREE MEMORY.
                                ;
                                ;INPUTS:
                                ;       R1 - FILE TYPE
                                ;               1 - UDA52 TEST 1 DM PROGRAM
                                ;               2 - UDA50 TEST 1 DM PROGRAM
                                ;               3 - UDA52 TEST 2 DM PROGRAM
                                ;               4 - UDA50 TEST 2 DM PROGRAM
                                ;               5 - UDA52 TEST 3 DM PROGRAM
                                ;               6 - UDA50 TEST 3 DM PROGRAM
                                ;               7 - TEST 4 QUESTIONS
                                ;               8 - UDA52 TEST 4 DM PROGRAM
                                ;               9 - UDA50 TEST 4 DM PROGRAM
                                ;               10 - DRIVE DIAGNOSTIC DOWNLINE LOAD PROGRAM
                                ;       DLLNAM - IF R1 CONTAINS 10, TWO WORDS AT THIS ADDRESS CONTAIN
                                ;                NAME OF PROGRAM IN RAD50.
                                ;       R5 - ADJUSTED ADDRESS WHERE TO BRING DATA INTO.
                                ;OUTPUTS:
                                ;       DATA FROM RECORD IN MEMORY
                                ;       CARRY CLEAR IF NO ERROR, CARRY SET IF ERROR

107044                          RDREC:  PUSH <R0,R1,R2,R3,R4,R5>
107044  010046                                                          MOV R0,-(SP)
107046  010146                                                          MOV R1,-(SP)
107050  010246                                                          MOV R2,-(SP)
107052  010346                                                          MOV R3,-(SP)
107054  010446                                                          MOV R4,-(SP)
107056  010546                                                          MOV R5,-(SP)
107060  005037  065046                  CLR     FNUM
107064  005737  065100                  TST FILOPN                      ;SEE IF FILE ALREADY OPEN
107070  001005                          BNE RDSTS
107072                                  OPEN #FNAME                     ;IF NOT, OPEN FILE NOW
107072  012700  065062                                                  MOV     #FNAME,R0
107076  104434                                                          TRAP    C$OPEN
107100  005237  065100                  INC FILOPN                      ;AND MARK AS OPEN
107104  005105                  RDSTS:  COM R5                          ;COMPLEMENT LOAD ADDRESS (SEARCH MODE)
107106                          RDST:   BREAK                           ;ALLOW PROGRAM TO BE INTERRUPTED
107106  104422                                                          TRAP    C$BRK
                                                                        ;GETBYTE CALLS DON'T SEEM TO BREAK ON CONTROL-C!
107110                                  GETBYTE R4                      ;GET A BYTE
107110  104426                                                          TRAP    C$GETB
107112  110004                                                          MOVB    R0,R4
107114  005704                          TST R4                          ;IF ZERO
107116  001773                          BEQ RDST                        ; KEEP READING
107120  022704  000001                  CMP #1,R4                       ;WHEN NOT ZERO
107124  001142                          BNE RWRDE1                      ; IT BETTER BE A ONE
107126                                  GETBYTE R0                      ; AND THE NEXT BYTE
107126  104426                                                          TRAP    C$GETB
107130  060004                          ADD R0,R4
107132  005700                          TST R0                          ; IF ZERO, PROCESS DATA
107134  001431                          BEQ RDDAT
```

```
107136  020001                          CMP R0,R1               ;CHECK IF TYPE OF FILE LOOKING FOR
107140  103427                          BLO RDDAT               ;IF TOO SOON IN FILE, KEEP SEARCHING
107142  101121                          BHI RDERR               ;IF PAST TYPE, GIVE ERROR RETURN
107144  004737  107364                  CALL FWORD              ;GET NEXT TWO WORDS
107150  013702  065076                  MOV FDATA,R2
107154  004737  107364                  CALL FWORD
107160                                  GETBYTE R0              ;GET CHECKSUM
107160  104426                                                          TRAP    C$GETB
107162  060004                          ADD R0,R4               ;ADD TO COMPUTED SUM
107164  105704                          TSTB R4                 ;SEE IF THIS SUM IS ZERO
107166  001121                          BNE RWRDE1              ;  IF NOT, REPORT CHECKSUM ERROR
107170  020127  000012                  CMP R1,#10.             ;IF FILE TYPE IS A 10
107174  001007                          BNE 1$
107176  023702  065300                  CMP DLLNAM,R2           ;  MATCH THE PROGRAM NAME
107202  001341                          BNE RDST                ;KEEP SEARCHING IF NOT DESIRED PROGRAM
107204  023737  065302  065076          CMP DLLNAM+2,FDATA
107212  001335                          BNE RDST
107214  005105                  1$:     COM R5                  ;GET STORAGE ADDRESS
107216  000733                          BR RDST                 ;SWITCH FROM SEARCH TO STORE MODE
```

```
      1 107220  004737  107364  RDDAT:  CALL FWORD                ;READ BYTE COUNT
      2 107224  013703  065076          MOV FDATA,R3              ;SAVE IN R3
      3 107230  004737  107364          CALL FWORD                ;READ LOAD ADDRESS
      4 107234  162703  000006          SUB #6,R3                 ;SUBTRACT BYTES ALREADY READ FROM BYTE COUNT
      5 107240  001431                  BEQ RWORDT                ;IF RESULT IS ZERO, THIS IS A
      6                                                           ;TRANSFER BLOCK
      7 107242  005705                  TST R5                    ;IF IN SEARCH MODE,
      8 107244  100413                  BMI 3$                    ;  BYPASS TRANSFER ADDRESS COMPUTATION
      9 107246  013701  065076          MOV FDATA,R1              ;GET LOAD ADDRESS
     10 107252  060501                  ADD     R5,R1             ; R1 -> REAL STARTING ADDRESS
     11 107254  020127  002122          CMP     R1,#STORAG        ; R1 MUST BE GREATER THAN STORAG
     12 107260  103452                  BLO     RDERR             ; IF NOT, ERROR
     13 107262  060301                  ADD R3,R1                 ;ADD BYTES IN RECORD
     14 107264  022701  064742          CMP     #<STORAG+STOSIZ>,R1 ; R1 MUST BE LESS THAN ENDING ADDRESS
     15 107270  103446                  BLO     RDERR             ; IF NOT, ERROR
     16 107272  160301                  SUB     R3,R1
     17 107274                  3$:     GETBYTE R0                ;GET DATA BYTE
        107274  104426                                                      TRAP    C$GETB
     18 107276  005705                  TST R5                    ;IF IN SEARCH MODE,
     19 107300  100401                  BMI 4$                    ;  BYPASS DATA STORAGE
     20 107302  110021                  MOVB R0,(R1)+             ;STORE IN MEMORY
     21 107304  060004          4$:     ADD R0,R4                 ;UPDATE CHECKSUM
     22 107306  005303                  DEC R3                    ;COUNT THE BYTE
     23 107310  001371                  BNE 3$                    ;GET THEM ALL
     24 107312                          GETBYTE R0                ;GET CHECKSUM
        107312  104426                                                      TRAP    C$GETB
     25 107314  060004                  ADD R0,R4                 ;ADD
     26 107316  105704                  TSTB R4                   ;IF CHECKSUM CORRECT,
     27 107320  001672                  BEQ RDST                  ;  THEN GO READ NEXT RECORD
     28 107322  000443                  BR RWRDE1                 ;  ELSE REPORT ERROR
```

```
      1 107324                          RWORDT: GETBYTE R0                      ;READ CHECKSUM BYTE
        107324  104426                                                                  TRAP    C$GETB
      2 107326  060004                          ADD R0,R4                       ;ADD TO COMPUTED CHECKSUM
      3 107330  105704                          TSTB R4                         ;CHECK LOW BYTE OF SUM
      4 107332  001037                          BNE RWRDE1                      ;BRANCH IF CHECKSUM ERROR
      5 107334  005705                          TST R5                          ;IF IN SEARCH MODE,
      6 107336  100663                          BMI RDST                        ;  KEEP ON SEARCHING
      7 107340                                  POP <R5,R4,R3,R2,R1,R0>
        107340  012605                                                                  MOV (SP)+,R5
        107342  012604                                                                  MOV (SP)+,R4
        107344  012603                                                                  MOV (SP)+,R3
        107346  012602                                                                  MOV (SP)+,R2
        107350  012601                                                                  MOV (SP)+,R1
        107352  012600                                                                  MOV (SP)+,R0
      8 107354  010137  065046                  MOV     R1,FNUM
      9 107360  000241                          CLC
     10 107362  000207                          RETURN
     11
     12 107364                          FWORD:  GETBYTE R0                      ;READ A BYTE FROM FILE
        107364  104426                                                                  TRAP    C$GETB
     13 107366  060004                          ADD R0,R4                       ;UPDATE CHECKSUM ERROR
     14 107370  110037  065076                  MOVB R0,FDATA                   ;START TO BUILD WORD
     15 107374                                  GETBYTE R0                      ;READ ANOTHER BYTE FROM FILE
        107374  104426                                                                  TRAP    C$GETB
     16 107376  060004                          ADD R0,R4                       ;UPDATE CHECKSUM
     17 107400  110037  065077                  MOVB R0,FDATA+1                 ;COMPLETE WORD
     18 107404  000207                          RETURN
```

```
      1 107406  004737  107026          RDERR:  CALL CLOSEF                     ;CLOSE FILE AS POSITION IS LOST
      2 107412                                  POP <R5,R4,R3,R2,R1,R0>
        107412  012605                                                           MOV (SP)+,R5
        107414  012604                                                           MOV (SP)+,R4
        107416  012603                                                           MOV (SP)+,R3
        107420  012602                                                           MOV (SP)+,R2
        107422  012601                                                           MOV (SP)+,R1
        107424  012600                                                           MOV (SP)+,R0
      3 107426  000261                          SEC                             ;ERROR RETURN, FILE NOT FOUND
      4 107430  000207                          RETURN
      5
      6 107432                          RWRDE1: ERRSF 5,,ERR005
        107432  104454                                                           TRAP    C$ERSF
        107434  000005                                                           .WORD   5
        107436  000000                                                           .WORD   0
        107440  074706                                                           .WORD   ERR005
      7 107442                                  DOCLN
        107442  104444                                                           TRAP    C$DCLN
```

```
                                        ;KW11I
                                        ;
                                        ;CLOCK INTERRUPT SERVICE ROUTINE
                                        ;
107444                                  BGNSRV KW11I
107444                                                                  KW11I::
107444  062737  000001  065232                ADD #1,KW.EL            ;COUNT THE INTERRUPT
107452  005537  065234                      ADC KW.EL+2
107456  012777  000105  155536              MOV #KWOUT.,@KW.CSR       ;RESTART THE CLOCK
107464                                  ENDSRV
107464                                                                  L10042:
107464  000002                                                                  RTI

107466                                  BGNSRV  INTSRV                  ; UDA INTERRUPT SERVER
107466                                                                  INTSRV::
107466  005237  065060                  INC     INTRCV                  ; FLAG INTERRUPT AS RECEIVED
107472                                  ENDSRV
107472                                                                  L10043:
107472  000002                                                                  RTI
```

```
                                ;RESET
                                ; RESET ALL UDA-50S IN THE CONTROLLER TABLES
                                ;
                                ; INPUTS:
                                ;       IPADRS - CONTAINS ALL IP ADDRESSES
                                ; OUTPUTS:
                                ;       NONE
                                ;
107474                          RESET:  PUSH <R3,R4>
107474  010346                                                          MOV R3,-(SP)
107476  010446                                                          MOV R4,-(SP)
107500  005037  065242                CLR NXMAD               ; CLEAR NON-EXISTANT MEMORY ADDRESS
107504                                  SETVEC #4,#NXMI,#PRI07
107504  012746  000340                                                  MOV     #PRI07,-(SP)
107510  012746  105716                                                  MOV     #NXMI,-(SP)
107514  012746  000004                                                  MOV     #4,-(SP)
107520  012746  000003                                                  MOV     #3,-(SP)
107524  104437                                                          TRAP    C$SVEC
107526  062706  000010                                                  ADD     #10,SP
107532                                  BREAK
107532  104422                                                          TRAP    C$BRK
107534  012703  000010                MOV     #8.,R3          ; R3 = COUNTER OF ENTRIES
107540  012704  065140                MOV     #IPADRS,R4      ; R4 -> IP ADDRESS
107544  005714                  1$:     TST     (R4)            ; IS THERE AN ENTRY?
107546  001406                          BEQ     2$              ; IF NOT, DONE
107550  005034                          CLR     @(R4)+          ; INIT UDA
107552  005737  065242                TST     NXMAD           ; WAS THERE AN ERROR?
107556  001005                          BNE     3$              ; IF SO, BRANCH
107560  005303                          DEC     R3              ; MAKE SURE WE DO NOT EXTEND OVER AREA
107562  001370                          BNE     1$              ; IF NOT DONE, BRANCH
107564                          2$:     POP     <R4,R3>
107564  012604                                                          MOV (SP)+,R4
107566  012603                                                          MOV (SP)+,R3
107570  000207                          RETURN
                                ;
107572  005744                  3$:     TST     -(R4)           ; R4 -> UDAIP IN ERROR
107574  010405                          MOV     R4,R5           ; R5 -> UDAIP
107576                                  ERRDF   20,,ERR020
107576  104455                                                          TRAP    C$ERDF
107600  000024                                                          .WORD   20
107602  000000                                                          .WORD   0
107604  075006                                                          .WORD   ERR020
107606  005014                          CLR     (R4)            ; DESTROY ADDRESS SO NOT TO FALL IN ENDLESS RESET ERROR LOOP
107610                                  DORPT
107610  104424                                                          TRAP    C$DRPT
107612                                  DOCLN
107612  104444                                                          TRAP    C$DCLN
```

```
                                        ;RNTIME
                                        ;
                                        ;PRINT RUNTIME
                                        ;
                                        ;INPUTS:
                                        ;       KW.EL - CONTAINS ELAPSED TIME
                                        ;       KW.HZ - HERTZ OF CLOCK
                                        ;OUTPUTS:
                                        ;       IF CLOCK ON SYSTEM:
                                        ;          '' RNTIME HH:MM:SS '' PRINTED
                                        ;       IF NO CLOCK: ONE SPACE IS PRINTED
                                        ;
107614  005737  065222        RNTIME: TST KW.CSR                      ;CHECK IF A CLOCK PRESENT
107620  001465                  BEQ RNTIMX                      ;BRANCH IF NOT
107622                          PUSH <R0,R3,R4,R5>
107622  010046                                                          MOV R0,-(SP)
107624  010346                                                          MOV R3,-(SP)
107626  010446                                                          MOV R4,-(SP)
107630  010546                                                          MOV R5,-(SP)
107632  013703  065232          MOV KW.EL,R3                    ;GET ELAPSED TIME
107636  013704  065234          MOV KW.EL+2,R4
107642  013700  065230          MOV KW.HZ,R0                    ;GET SPEED OF CLOCK
107646  004737  103542          CALL DIVIDE                     ;COMPUTE SECONDS OF ELAPSED TIME
107652  012700  000074          MOV #60.,R0                     ;NOW DIVIDE BY 60
107656  004737  103542          CALL DIVIDE                     ; TO COMPUTE MINUTES
107662                          PUSH R5                         ;SAVE REMAINDER AS SECONDS
107662  010546                                                          MOV R5,-(SP)
107664  004737  103542          CALL DIVIDE                     ;DIVIDE BY 60 AGAIN
107670                          PNT RNTIM,R3                    ;PRINT HOURS
107670  010346                                                          MOV R3,-(SP)
107672  004137  104250                                                  JSR R1,LPNT
107676  065451                                                          .WORD RNTIM
107700  000002                                                          .WORD PNT.CT
107702  020527  000011          CMP R5,#9.                      ;IF MINUTES 9 OR LESS
107706  003004                  BGT 1$
107710                          PRINT #'0                       ;PRINT A LEADING ZERO
107710  112700  000060                                                  MOVB #'0,R0
107714  004737  104040                                                  CALL CPNT
107720                  1$:     PNT RNTIM1,R5                   ;NOW PRINT MINUTES
107720  010546                                                          MOV R5,-(SP)
107722  004137  104250                                                  JSR R1,LPNT
107726  065474                                                          .WORD RNTIM1
107730  000002                                                          .WORD PNT.CT
107732                          POP R5                          ;GET SECONDS
107732  012605                                                          MOV (SP)+,R5
107734  020527  000011          CMP R5,#9.                      ;IF 9 OR LESS
107740  003004                  BGT 2$
107742                          PRINT #'0                       ;PRINT A LEADING ZERO
107742  112700  000060                                                  MOVB #'0,R0
107746  004737  104040                                                  CALL CPNT
107752                  2$:     PNT RNTIM2,R5                   ;NOW PRINT SECONDS
107752  010546                                                          MOV R5,-(SP)
107754  004137  104250                                                  JSR R1,LPNT
107760  065502                                                          .WORD RNTIM2
107762  000002                                                          .WORD PNT.CT
107764                          POP <R5,R4,R3,R0>               ;HOURS IN R3
107764  012605                                                          MOV (SP)+,R5
```

```
     107766  012604                                                  MOV (SP)+,R4
     107770  012603                                                  MOV (SP)+,R3
     107772  012600                                                  MOV (SP)+,R0
  35 107774                        RNTIMX: PRINT <#' >                ;PRINT A SPACE
     107774  112700  000040                                          MOVB #' ,R0
     110000  004737  104040                                          CALL CPNT
  36 110004  000207                        RETURN
  37
  38 110006                        ENDMOD
```

```
                                .SBTTL  REPORT CODING SECTION

110006                                  BGNMOD

                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--

110006                                  BGNRPT
110006                                                                  L$RPT::

110006                                  PUSH    <R0,R1,R2,R3,R4,R5>
110006  010046                                                                  MOV R0,-(SP)
110010  010146                                                                  MOV R1,-(SP)
110012  010246                                                                  MOV R2,-(SP)
110014  010346                                                                  MOV R3,-(SP)
110016  010446                                                                  MOV R4,-(SP)
110020  010546                                                                  MOV R5,-(SP)
110022                                  PNTS RPTMSG,TNUM                ;PRINT TEST NUMBER
110022  013746  065050                                                          MOV TNUM,-(SP)
110026  004137  104242                                                          JSR R1,LPNTS
110032  110462                                                                  .WORD RPTMSG
110034  000002                                                                  .WORD PNT.CT
110036  004737  107614                  CALL RNTIME                     ;GET RUNTIME PARAMETERS
110042                                  PRINT #CR                       ;END THE LINE
110042  112700  000015                                                          MOVB #CR,R0
110046  004737  104040                                                          CALL CPNT
110052  012701  065236                  MOV #STIME,R1                   ;AT 15 MINUTES FROM NOW
110056  012700  001604                  MOV #15.*60.,R0                 ;SET TIME FOR NEXT REPORT
110062  004737  105736                  CALL SETTO
```

REPORT CODING SECTION

```
     4 110066  023727  065050  000004          CMP TNUM,#4                     ;IF NOT TEST 4
     6 110074  001402                          BEQ 1$                          ;
     7 110076  000137  110374                  JMP RPTXX                       ;  THAT IS ALL
     8 110102                          1$:     PNTS RPTMSH
       110102  004137  104242                                                          JSR R1,LPNTS
       110106  110516                                                                  .WORD RPTMSH
       110110  000000                                                                  .WORD PNT.CT
     9 110112  013705  065026                  MOV CTABS,R5                    ;GET ADDRESS OF FIRST CONTROLLER TABLE
    10 110116                          RPTCT:
    11 110116  005765  000002                  TST C.UNIT(R5)                  ;SEE IF CONTROLLER AVAILABLE FOR TESTING
    13 110122  100520                          BMI RPTCTN
    14 110124                                        ASSUME CT.AVL EQ BIT15
    20 110124  010504                          MOV R5,R4                       ;COMPUTE ADDRESS OF DRIVE TABLE POINTERS
    21 110126  062704  000020                  ADD #C.DRQ,R4
    22 110132  012703  000010                  MOV #8.,R3                      ;GET COUNT OF DRIVES
    23 110136  012401                  RPTDT:  MOV (R4)+,R1                    ;LOOK AT POINTER
    24 110140  001511                          BEQ RPTCTN                      ;GO TO NEXT IF NO TABLE
    25 110142  005761  000002                  TST D.UNIT(R1)                  ;SEE IF DRIVE AVAILABLE
    27 110146  100504                          BMI RPTDTN
    28 110150                                        ASSUME DT.AVL EQ BIT15
```

REPORT CODING SECTION

```
      1 110150                                  PUSH <R3,R4,R5,R1>
        110150  010346                                                 MOV R3,-(SP)
        110152  010446                                                 MOV R4,-(SP)
        110154  010546                                                 MOV R5,-(SP)
        110156  010146                                                 MOV R1,-(SP)
      2 110160  012700  065102                  MOV #TEMP,R0          ;PLACE 18 SPACE CHARACTERS INTO
      3 110164  012701  000022                  MOV #18.,R1           ;  TEMP STORAGE
      4 110170  112720  000040        1$:       MOVB #' ,(R0)+
      5 110174  005301                          DEC R1
      6 110176  001374                          BNE 1$
      7 110200  005010                          CLR (R0)              ;THEN A NULL CHARACTER
      8 110202  011605                          MOV (SP),R5           ;GET DRIVE TABLE STORAGE ADDRESS
      9 110204  016501  000200                  MOV D.SERN(R5),R1     ;GET SERIAL NUMBER
     10 110210  016502  000202                  MOV D.SERN+2(R5),R2
     11 110214  016503  000204                  MOV D.SERN+4(R5),R3
     12 110220  005004                          CLR R4
     13 110222  004737  110414        2$:       CALL DIV10            ;DIVIDE BY 10
     14 110226  062705  000060                  ADD #'0,R5            ;CONVERT TO ASCII CHARACTER
     15 110232  110540                          MOVB R5,-(R0)         ;PUT DIGIT INTO TEMP STORAGE
     16 110234  010146                          MOV R1,-(SP)
     17 110236  050216                          BIS R2,(SP)           ;SEE IF QUOTIENT IS ZERO
     18 110240  050316                          BIS R3,(SP)
     19 110242  050426                          BIS R4,(SP)+
     20 110244  001366                          BNE 2$                ;IF NOT, DIVIDE AGAIN
     21 110246                                  POP R1
        110246  012601                                                 MOV (SP)+,R1
     22 110250                                  PRINTS #RPTMSD,D.UNIT(R1),(R1),#TEMP,D.SEEK(R1),D.XFRR(R1),D.XFRW(R1)
        110250  016146  000164                                         MOV     D.XFRW(R1),-(SP)
        110254  016146  000166                                         MOV     D.XFRR(R1),-(SP)
        110260  016146  000174                                         MOV     D.SEEK(R1),-(SP)
        110264  012746  065102                                         MOV     #TEMP,-(SP)
        110270  011146                                                 MOV     (R1),-(SP)
        110272  016146  000002                                         MOV     D.UNIT(R1),-(SP)
        110276  012746  110734                                         MOV     #RPTMSD,-(SP)
        110302  012746  000007                                         MOV     #7,-(SP)
        110306  010600                                                 MOV     SP,R0
        110310  104416                                                 TRAP    C$PNTS
        110312  062706  000020                                         ADD     #20,SP
     23 110316                                        ASSUME D.DRV EQ 0
     24 110316                                  PRINTS #RPTMD2,D.HERR(R1),D.SERR(R1),D.ECCC(R1)
        110316  016146  000176                                         MOV     D.ECCC(R1),-(SP)
        110322  016146  000172                                         MOV     D.SERR(R1),-(SP)
        110326  016146  000170                                         MOV     D.HERR(R1),-(SP)
        110332  012746  111003                                         MOV     #RPTMD2,-(SP)
        110336  012746  000004                                         MOV     #4,-(SP)
        110342  010600                                                 MOV     SP,R0
        110344  104416                                                 TRAP    C$PNTS
        110346  062706  000012                                         ADD     #12,SP
     48 110352                                  POP <R5,R4,R3>
        110352  012605                                                 MOV (SP)+,R5
        110354  012604                                                 MOV (SP)+,R4
        110356  012603                                                 MOV (SP)+,R3
```

```
      1 110360  005303                RPTDTN: DEC R3                          ;COUNT THE DRIVE TABLES
      2 110362  003265                        BGT RPTDT                       ;REPEAT FOR ALL DRIVE TABLES
      3 110364  062705  000046        RPTCTN: ADD #C.SIZE,R5                  ;GO TO NEXT CONTROLLER TABLE
      4 110370  005715                        TST (R5)
      9 110372  001251                        BNE RPTCT
     11 110374                        RPTXX:  POP     <R5,R4,R3,R2,R1,R0>
        110374  012605                                                                  MOV (SP)+,R5
        110376  012604                                                                  MOV (SP)+,R4
        110400  012603                                                                  MOV (SP)+,R3
        110402  012602                                                                  MOV (SP)+,R2
        110404  012601                                                                  MOV (SP)+,R1
        110406  012600                                                                  MOV (SP)+,R0
     12 110410                                EXIT    RPT
        110410  000167                                                                  .WORD   J$JMP
        110412  000412                                                                  .WORD   L10044-2-.
     13
     14 110414                        DIV10:  PUSH R0                         ;DIVIDEND IS IN <R4,R3,R2,R1>
        110414  010046                                                                  MOV R0,-(SP)
     15 110416  012700  000100                MOV #64.,R0                     ;SET UP SHIFT COUNT
     16 110422  005005                        CLR R5                          ;START WITH ZERO REMAINDER
     17 110424  006301                1$:     ASL R1
     18 110426  006102                        ROL R2                          ;SHIFT LEFT INTO R5
     19 110430  006103                        ROL R3
     20 110432  006104                        ROL R4
     21 110434  006105                        ROL R5
     22 110436  022705  000012                CMP #10.,R5                     ;SILL DIVISOR GO INTO REMAINDER?
     23 110442  101003                        BHI 2$                          ;ONLY SUBTRACT IF IT WILL
     24 110444  162705  000012                SUB #10.,R5                     ;SUBTRACT DIVISOR
     25 110450  005201                        INC R1                          ;PUT A ONE INTO QUOTIENT
     26 110452  005300                2$:     DEC R0                          ;COUNT THE SHIFTS
     27 110454  001363                        BNE 1$
     28 110456                                POP R0                          ;RETURN WITH QUOTIENT IN
        110456  012600                                                                  MOV (SP)+,R0
     29 110460  000207                        RETURN                          ;  <R4,R3,R2,R1> AND REMAINDER IN R5
     30
     31 110462      116       042      124  RPTMSG: .ASCIZ\N''TEST ''D3'' IN PROGRESS.  ''\
     32 110516      116       042      125  RPTMSH: .ASCII\N''UNIT DRIVE      SERIAL-NUMBER SEEKS MBYTES  MBYTES  HARD   SOFT    ECC''N\
     33 110630      042       040      040          .ASCIZ \''                                    X1000  READ  WRITTEN ERRORS ERRORS''N\
     34 110734      045       123      062  RPTMSD: .ASCIZ\%S2%D2%S3%D3%S1%T%S1%D5%S2%D5%S3%D5%S2\
     35 111003      045       104      065  RPTMD2: .ASCIZ\%D5%S2%D5%S1%D5%N\
     41                                           .EVEN
     42
     43 111026                                ENDRPT
        111026                                                                  L10044:
        111026  104425                                                                    TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

                                ;++
                                ; THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ; TO PROTECT THE LOAD MEDIA.
                                ;--

111030                                  BGNPROT
111030                                                                  L$PROT::

111030  177777                          -1                      ;OFFSET INTO P-TABLE FOR CSR ADDRESS
111032  177777                          -1                      ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
111034  177777                          -1                      ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

111036                                  ENDPROT
```

```
                        .SBTTL  INITIALIZE SECTION

                        ;++
                        ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                        ; AT THE BEGINNING OF EACH PASS.
                        ;--

111036                          BGNINIT
111036                                                          L$INIT::

                        .REM &
                            IF HERE FROM CONTINUE COMMAND
                              THEN
                                SET ICONT BIT IN IFLAGS
                            ENDIF
                            IF HERE FROM RESTART COMMAND
                              THEN
                                SET IREST BIT IN IFLAGS
                            ENDIF
                            IF HERE FROM POWER FAIL RESTART
                              THEN
                                RESET ALL UNITS
                                PRINT STATISTICAL REPORT
                            ENDIF
                            IF HERE FROM START COMMAND
                              THEN
                                RESET ALL UNITS
                                ESTABLISH FREE MEMORY
                                CLEAR TNUM
                                SET ISTRT BIT IN IFLAGS, CLEAR OTHER BITS
                                INITIALIZE CLOCK
                                BUILD TABLES
                            ELSE
                                CHECK TABLES FOR ADDED OR DROPPED UNITS
                            .ENDF
                        &
```

```
 1 111036                                      READEF #EF.CONTINUE          ;HERE FROM CONTINUE COMMAND?
   111036  012700  000036                                                                MOV    #EF.CONTINUE,RO
   111042  104447                                                                        TRAP   C$REFG
 2 111044                                      BNCOMPLETE INIT1             ;JUMP IF NOT
   111044  103006                                                                        BCC    INIT1
 3 111046  042737  000020  065044              BIC #ISTRTH,IFLAGS
 4 111054  052737  000002  065044              BIS #ICONT,IFLAGS            ;SET CONTINUE BIT IN FLAG RECORD
 5 111062                              INIT1:  READEF #EF.RESTART           ;LOOK AT EVENT FLAGS
   111062  012700  000037                                                                MOV    #EF.RESTART,RO
   111066  104447                                                                        TRAP   C$REFG
 6 111070                                      BNCOMPLETE INIT1A            ;SET IREST BIT IN IFLAGS
   111070  103005                                                                        BCC    INIT1A
 7 111072  052737  000004  065044              BIS #IREST,IFLAGS            ;  IF HERE FROM RESTART COMMAND
 8 111100  004737  107474                      CALL RESET                   ; RESET UDA-50'S
 9 111104                              INIT1A: READEF #EF.PWR               ;HERE FROM POWER RESTART?
   111104  012700  000034                                                                MOV    #EF.PWR,RO
   111110  104447                                                                        TRAP   C$REFG
10 111112                                      BNCOMPLETE INIT2             ;JUMP IF SET
   111112  103003                                                                        BCC    INIT2
11 111114  004737  107474                      CALL RESET                   ;RESET ALL UNITS
12 111120                                      DORPT                        ;PRINT A STATISTICAL REPORT
   111120  104424                                                                        TRAP   C$DRPT
13 111122                              INIT2:  READEF #EF.START             ;HERE FROM START COMMAND?
   111122  012700  000040                                                                MOV    #EF.START,RO
   111126  104447                                                                        TRAP   C$REFG
14 111130                                      BCOMPLETE INIT3              ;JUMP IF SO
   111130  103467                                                                        BCS    INIT3
```

```
                                        ;SET NOT AVAILABLE BITS IN ALL CONTROLLER AND DRIVE TABLES
                                
111132  013705  065026                        MOV CTABS,R5                    ;GET FIRST CONTROLLER TABLE ADDRESS
                                
111136  052765  100000  000002  INITC1: BIS #CT.AVL,C.UNIT(R5)                 ;SET BIT IN CONTROLLER TABLE
111144  010502                        MOV R5,R2                               ;GET POINTER TO DRIVE TABLES
111146  062702  000020                ADD #C.DRO,R2
111152  012703  000010                MOV #8.,R3                              ;GET COUNT OF DRIVE TABLES
111156  012200                INITC2: MOV (R2)+,R0                            ;CHECK IF ANY MORE DRIVE TABLES
111160  001405                        BEQ INITC3
111162  052760  100000  000002        BIS #DT.AVL,D.UNIT(R0)                  ;SET BIT IN DRIVE TABLE
111170  005303                        DEC R3
111172  003371                        BGT INITC2
111174  062705  000046        INITC3: ADD #C.SIZE,R5                          ;MOVE TO NEXT CONTROLLER TABLE
111200  005715                        TST (R5)                                ;IS THERE A NEXT ONE?
111202  001355                        BNE INITC1                              ;IF SO, CLEAR THE BITS THERE
                                
                                        ;NOW GET EACH P-TABLE AND CLEAR NOT AVAILABLE BITS
                                
111204  005003                        CLR R3                                  ;START WITH UNIT 0
111206                          INITC4: GPHARD R3,R0                            ;GET HW P-TABLE
111206  010300                                                                        MOV     R3,R0
111210  104442                                                                      TRAP    C$GPHRD
                                
111212                                  BNCOMPLETE INITC7                       ;GO AROUND IF NOT AVAILABLE
111212  103030                                                                      BCC     INITC7
111214  013705  065026                MOV CTABS,R5                            ;GET FIRST CONTROLLER TABLE
111220  021015                INITC5: CMP (R0),(R5)                           ;COMPARE UDA ADDRESSES
111222  001411                        BEQ INITCC
111224  062705  000046                ADD #C.SIZE,R5                          ;LOOK AT NEXT CONTROLLER TABLE
111230  005715                        TST (R5)                                ;IF THERE IS ANY
111232  001372                        BNE INITC5
111234                          INITE1: ERRSF 6,,ERR006
111234  104454                                                                      TRAP    C$ERSF
111236  000006                                                                      .WORD   6
111240  000000                                                                      .WORD   0
111242  074744                                                                      .WORD   ERR006
                                
111244                                  DOCLN
111244  104444                                                                      TRAP    C$DCLN
111246  016001  000010          INITCC: MOV HO.LDR(R0),R1
111252  004737  103132                CALL GTDRVT
111256  001366                        BNE INITE1
111260  042765  100000  000002  INITC6: BIC #CT.AVL,C.UNIT(R5)                 ;CLEAR BIT IN CONTROLLER TABLE
111266  042764  100000  000002        BIC #DT.AVL,D.UNIT(R4)                  ;CLEAR BIT IN DRIVE TABLE
111274  005203                INITC7: INC R3                                  ;INCREMENT UNIT NUMBER
111276  023703  002012                CMP L$UNIT,R3                           ;CHECK IF GOT ALL TABLES
111302  003341                        BGT INITC4                              ;IF NOT, GO GET ANOTHER
111304  000137  112354                JMP INITXX                              ;EXIT THE INIT CODE
```

```
      1 111310  004737  107474          INIT3:  CALL RESET                      ;RESET ALL UNITS
      2 111314                                  MEMORY FFREE                    ;RESET START OF FREE MEMORY
        111314  104431                                                                  TRAP    C$MEM
        111316  010037  065016                                                          MOV     R0,FFREE
      3 111322  017737  153470  065020          MOV @FFREE,FSIZE                ;RESET SIZE OF FREE MEMORY
      4 111330  005037  065050                  CLR TNUM                        ;INITIALIZE TEST NUMBER TO NO TEST RUNNING
      5 111334  005037  065046                  CLR FNUM                        ;INITIALIZE FILE NUMBER TO NO FILE IN MEMORY
      6 111340  012737  000010  065044          MOV #ISTRT,IFLAGS               ;SET START FLAG FOR TEST 4
      7
      8                                   ;INITIALIZE CLOCK
      9
     10            000105                        KWOUT.=105                      ;DATA TO SEND TO KW11 TO START CLOCK
     11 111346  005037  065232                  CLR KW.EL                       ;CLEAR ELAPSED TIME
     12 111352  005037  065234                  CLR KW.EL+2
     13 111356                                  CLOCK L,R0                      ;SEE IF AN L CLOCK PRESENT
        111356  012700  000114                                                          MOV     #'L,R0
        111362  104462                                                                  TRAP    C$CLCK
     14 111364                                  BCOMPLETE KYES
        111364  103413                                                                  BCS     KYES
     15 111366                                  CLOCK P,R0                      ;SEE IF A P CLOCK PRESENT
        111366  012700  000120                                                          MOV     #'P,R0
        111372  104462                                                                  TRAP    C$CLCK
     16 111374                                  BCOMPLETE KYES
        111374  103407                                                                  BCS     KYES
     17 111376  005037  065222                  CLR KW.CSR                      ;IF NEITHER, CLEAR CSR STORAGE WORD
     18 111402                                  PNTF NOCLOCK
        111402  004137  104212                                                          JSR R1,LPNTF
        111406  067072                                                                  .WORD NOCLOCK
        111410  000000                                                                  .WORD PNT.CT
     19 111412  000434                          BR KNO
     20 111414  012037  065222          KYES:   MOV (R0)+,KW.CSR                ;STORE DATA RETURNED
     21 111420  012037  065224                  MOV (R0)+,KW.BRL
     22 111424  012037  065226                  MOV (R0)+,KW.VEC
     23 111430  012037  065230                  MOV (R0)+,KW.HZ
     24 111434                                  SETVEC KW.VEC,#KW11I,#PRI07     ;KW.BRL ;SET THE VECTOR
        111434  012746  000340                                                          MOV     #PRI07,-(SP)
        111440  012746  107444                                                          MOV     #KW11I,-(SP)
        111444  013746  065226                                                          MOV     KW.VEC,-(SP)
        111450  012746  000003                                                          MOV     #3,-(SP)
        111454  104437                                                                  TRAP    C$SVEC
        111456  062706  000010                                                          ADD     #10,SP
     25 111462  012777  000105  153532          MOV #KWOUT.,@KW.CSR             ;START THE CLOCK
     26 111470  012701  065236                  MOV     #STIME,R1
     30 111474  012700  001604                  MOV #15.*60.,R0                 ;SET TIME FOR NEXT REPORT
     31 111500  004737  105736                  CALL SETTO
     33 111504                          KNO:
```

```
                                        ;INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS

111504  013737  065016  065026            MOV FFREE,CTABS                 ;STORE START OF CONTROLLER TABLES
111512  005077  153310                    CLR @CTABS                      ;ZEROS MARKS END CONTROLLER TABLES
111516  005037  065030                    CLR CTRLRS                      ;CLEAR CONTROLLER COUNT
111522  012701  065140                    MOV     #IPADRS,R1              ; R1 -> IP ADDRESS
111526  012702  000010                    MOV     #8.,R2                  ; R2 IS A COUNTER
111532  005021                    1$:     CLR     (R1)+                   ; CLEAR ENTRY
111534  005302                            DEC     R2                      ; DONE?
111536  001375                            BNE     1$                      ; IF NOT, BRANCH

                                        ;GET A P-TABLE FROM DRS

                                        ;LOGICAL UNIT NUMBER IN R2
                                INIT4:  GPHARD R2,R0                    ;GET POINTER TO A P-TABLE
111542  010200                                                                  MOV     R2,R0
111544  104442                                                                  TRAP    C$GPHRD
                                        BNCOMPLETE NXTTAB               ;IGNORE IF NO TABLE RETURNED
111546  103110                                                                  BCC     NXTTAB

                                        ;SEE IF A CONTROLLER TABLE ALREADY EXISTS FOR CONTROLLER IN P-TABLE

111550  013703  065026                    MOV CTABS,R3                    ;GET ADDRESS OF CONTROLLER TABLES
                                INIT5:  TST (R3)                        ;CHECK IF ANY MORE TABLES
111554  005713
111556  001416                            BEQ NEWTAB                      ;BUILD NEW TABLE IF FOUND ZERO WORD
111560  021013                            CMP (R0),(R3)                   ;CHECK IF SAME UNIBUS ADDRESS
                                                ASSUME C.UADR EQ 0
                                                ASSUME HO.UBA EQ 0
111562  001463                            BEQ SAMTAB                      ;CHECK TABLE IF ALREADY EXISTS
111564  016301  000004                    MOV     C.VEC(R3),R1            ;GET VECTOR FROM EXISING CONTROLLER TABLE
111570  042701  177000                    BIC     #^C<CT.VEC>,R1
111574  026001  000002                    CMP     HO.VEC(R0),R1           ;SEE IF DEFFERENT VECTOR
111600  001002                            BNE     1$
111602  000137  112502                    JMP     SAMVEC                  ;ERROR, CAN'T HAVE TWO UDA'S WITH SAME VECTOR
111606  062703  000046            1$:     ADD #C.SIZE,R3                  ;MOVE TO NEXT TABLE
111612  000760                            BR INIT5
```

```
                                        ;BUILD A CONTROLLER TABLE
                                        ;
111614  012703  000010    NEWTAB: MOV     #8.,R3          ;R3 IS A COUNTER
111620  012704  065140            MOV     #IPADRS,R4      ;R4 -> IP ADDRESSES
111624  005714            1$:     TST     (R4)            ; FOUND AN OPEN ENTRY?
111626  001404                    BEQ     2$              ; IF SO, GO FILL ENTRY
111630  005724                    TST     (R4)+           ; NEXT ENTRY
111632  005303                    DEC     R3              ; SEARCH THROUGH ENTIRE TABLE?
111634  001373                    BNE     1$              ; IF NOT, BRANCH
111636  000401                    BR      3$              ; ELSE, TABLE FULL
111640  011014            2$:     MOV     (R0),(R4)       ; STORE ENTRY INTO TABLE
111642  012701  000023    3$:     MOV #C.SIZE/2,R1        ;GET WORDS IN CONTROLLER TABLE
111646  004737  076034            CALL ALOCM              ;ALLOCATE SPACE FOR IT
111652  011021                    MOV (R0),(R1)+          ;STORE UNIBUS ADDRESS
111654  010221                    MOV R2,(R1)+            ;UNIT NUMBER
111656  016004  000004            MOV HO.BRL(R0),R4       ;GET BR LEVEL
111662  000304                    SWAB R4                 ;SWAP TO HIGH BYTE
111664  006104                    ROL R4                  ;SHIFT ONE MORE TO LEFT
111666  056004  000002            BIS HO.VEC(R0),R4       ;ADD VECTOR ADDRESS
111672  010421                    MOV R4,(R1)+            ;  TO TABLE
111674  016021  000006            MOV HO.BST(R0),(R1)+
111700  012721  004037            MOV #4037,(R1)+         ;PUT [JSR R0,UDASRV]
111704  012721  105726            MOV #UDASRV,(R1)+       ;  INTO TABLE
111710  012703  000015            MOV #13.,R3             ;CLEAR POINTERS TO DRIVE TABLES,
111714  005021            INIT7:  CLR (R1)+               ; TIMEOUT COUNTER, FLAGS, REF. NUMBER
111716  005303                    DEC R3
111720  001375                    BNE INIT7               ;LOOP TIL ALL CLEARED
111722  005237  065030            INC CTRLRS              ;COUNT THE CONTROLLER
111726  005011                    CLR (R1)                ;CLEAR TABLE END MARKER
111730  000417                    BR NXTTAB               ;NOW GO TO NEXT P-TABLE
```

```
                                        ;SHOULD BE SAME CONTROLLER, CHECK THAT OTHER PARAMETERS MATCH
                                        
111732  016004  000004        SAMTAB: MOV HO.BRL(R0),R4               ;GET BR LEVEL FROM P-TABLE
111736  000304                  SWAB R4                               ;SWAP TO HIGH BYTE
111740  006104                  ROL R4                                ;SHIFT ONE MORE TO LEFT
111742  056004  000002          BIS HO.VEC(R0),R4                     ;ADD VECTOR ADDRESS
111746  020463  000004          CMP R4,C.VEC(R3)                      ;COMPARE WITH CONTROLLER TABLE
111752  001004                  BNE 1$
111754  026063  000006  000006  CMP HO.BST(R0),C.BST(R3)              ;COMPARE BURST RATES
111762  001402                  BEQ NXTTAB
111764  000137  112432  1$:     JMP CTABER                            ;FATAL ERROR IF NOT SAME
                                        
                                        ;GET NEXT P-TABLE
                                        
111770  005202          NXTTAB: INC R2                                ;INCREMENT LOGICAL UNIT NUMBER
111772  023702  002012          CMP L$UNIT,R2                         ;CHECK IF GOT ALL TABLES
111776  003261                  BGT INIT4                             ;IF NOT, GO BACK FOR NEXT
                                        
112000  012701  000001          MOV #1,R1                             ;ALLOCATE SPACE FOR ZERO END WORD
112004  004737  076034          CALL ALOCM                            ;AFTER CONTROLLER TABLES
```

```
                                ;NOW BUILD DRIVE TABLES

                                        CLR R5                  ;CLEAR CUSTOMER DATA FLAG
                                        CLR R2                  ;LOGICAL UNIT NUMBER IN R2
                                INIT8:  GPHARD R2,R0            ;GET POINTER TO A P-TABLE
                                                                        MOV     R2,R0
                                                                        TRAP    C$GPHRD
                                        BNCOMPLETE INIT14       ;IF NOT AVAILABLE, GO GET NEXT
                                                                        BCC     INIT14

                                ;FIND CONTROLLER TABLE

                                        MOV CTABS,R3            ;GET ADDRESS OF CONTROLLER TABLES
                                INIT10: CMP (R0),(R3)           ;CHECK IF SAME UNIBUS ADDRESS
                                        BEQ INIT11              ;BRANCH IF TABLE FOUND
                                        ADD #C.SIZE,R3          ;MOVE TO NEXT TABLE
                                        BR INIT10
```

INITIALIZE SECTION

```
                                        ;BUILD DRIVE TABLE
                                        ;
112040  012701  000103          INIT11: MOV #D.SIZE/2,R1                ;GET SIZE OF DRIVE TABLE
112044  004737  076034                  CALL ALOCM                      ;ALLOCATE SPACE FROM FREE MEMORY
                                ;       R0 POINTS TO P-TABLE
                                ;       R1 POINTS TO DRIVE TABLE
                                ;       R3 POINTS TO CONTROLLER TABLE
                                ;       R2 IS UNIT NUMBER
112050  010337  065102                  MOV R3,TEMP                     ;SAVE CONTROLLER TABLE ADDRESS
                                                                        ;IN CASE AN ERROR IS DETECTED
112054  062703  000020                  ADD #C.DR0,R3                   ;BUILD POINTER TO C.DR ENTRY IN CONTROLLER TABLE
112060  012704  000010                  MOV #8.,R4                      ;GET MAX COUNT OF DRIVES ON ONE CONTROLLER
112064  005713                  INIT12: TST (R3)                        ;CHECK IF ENTRY CONTAINS POINTER TO DRIVE TABLE
112066  001411                          BEQ INIT13
112070  026033  000010                  CMP H0.LDR(R0),@(R3)+           ;CHECK DRIVE NUMBER IN DRIVE TABLE
112074  001002                          BNE 1$
112076  000137  112446                  JMP MLDRER                      ;IF SAME, TWO P-TABLES POINT TO SAME DRIVE
112102  005304                  1$:     DEC R4                          ;COUNT DRIVES
112104  001367                          BNE INIT12                      ;IF EIGHT DRIVE TABLES EXIST,
112106  000137  112464                  JMP TOOMER                      ;  THEN REPORT ERROR
112112  010113                  INIT13: MOV R1,(R3)                     ;LOAD DRIVE TABLE POINTER
112114  016021  000010                  MOV H0.LDR(R0),(R1)+            ;LOAD DRIVE NUMBER
112120  010221                          MOV R2,(R1)+                    ;LOAD UNIT NUMBER
112122  016011  000012                  MOV H0.PRM(R0),(R1)             ;GET TEST AREA BIT
112126  051105                          BIS (R1),R5                     ;SAVE "OR" OF BIT FROM ALL DRIVES
112130  005111                          COM (R1)                        ;COMPLIMENT IT
112132                                  AND HM.CYL,(R1)
112132  042711  157777                                                  BIC #^C<HM.CYL>,(R1)
112136  052721  011012                  BIS #DDEF,(R1)+                 ;LOAD DEFAULT PARAMETER BITS
112142  012703  000100                  MOV #<D.SIZE/2-3>,R3            ;CLEAR REST OF TABLE
112146  005021                  INIT3L: CLR (R1)+
112150  005303                          DEC R3
112152  003375                          BGT INIT3L
112154  012761  177777  177754         MOV #-1,<D.ECYL+2-D.SIZE>(R1)   ;MARK CYLINDERS AT TEST ALL
```

```
                                ;GO TO NEXT DRIVE TABLE
                                ;
112162  005202                  INIT14: INC R2                          ;INCREMENT LOGICAL UNIT NUMBER
112164  023702  002012                  CMP L$UNIT,R2                   ;CHECK IF GOT ALL TABLES
112170  003311                          BGT INIT8                       ;IF NOT, GET NEXT TABLE
                                ;
                                ;IF ANY DRIVE SELECTED FOR EXERCISE IN CUSTOMER DATA AREA
                                ;GIVE WARNING
                                ;
112172  032705  020000                  BIT #HM.CYL,R5                  ;CHECK IF BIT EVER SET
112176  001460                          BEQ INIT15                      ;BYPASS IF NOT
112200                                  PNTF INITWA                     ;PRINT WARNING HEADER
112200  004137  104212                                                          JSR R1,LPNTF
112204  066060                                                                  .WORD INITWA
112206  000000                                                                  .WORD PNT.CT
112210  013705  065026                  MOV CTABS,R5                    ;GET FIRST CONTROLLER TABLE
112214  010504          INITW1: MOV R5,R4                               ;GET ADDRESS OF POINTER TO DRIVE TABLE
112216  062704  000020                  ADD #C.DRO,R4
112222  012701  000010                  MOV #8.,R1                      ;GET COUNT OF DRIVE TABLES
112226  012403          INITW2: MOV (R4)+,R3                            ;GET ADDRESS OF DRIVE TABLE
112230  001422                          BEQ INITW4
112232  032763  020000  000004          BIT #D.DCY,D.PRM(R3)            ;CHECK IF CUSTOMER DATA SELECTED
112240  001014                          BNE INITW3
112242                                  PRINTF #INITWB,D.UNIT(R3),(R5),(R3)     ;PRINT NUMBERS
112242  011346                                                                  MOV     (R3),-(SP)
112244  011546                                                                  MOV     (R5),-(SP)
112246  016346  000002                                                          MOV     D.UNIT(R3),-(SP)
112252  012746  066164                                                          MOV     #INITWB,-(SP)
112256  012746  000004                                                          MOV     #4,-(SP)
112262  010600                                                                  MOV     SP,R0
112264  104417                                                                  TRAP    C$PNTF
112266  062706  000012                                                          ADD     #12,SP
112272  005301          INITW3: DEC R1                                  ;COUNT THE DRIVE TABLES
112274  001354                  BNE INITW2                              ;LOOK AT ALL OF THEM
112276  062705  000046  INITW4: ADD #C.SIZE,R5                         ;MOVE TO NEXT CONTROLLER TABLE
112302  005715                  TST (R5)                                ;SEE IF ANOTHER TABLE AND
112304  001343                  BNE INITW1                              ;  LOOK AT IT
```

INITIALIZE SECTION

```
                                ;GET CONFIRMATION TO PROCEED

112306                                  MANUAL                          ;CHECK IF MANUAL INTERVENTION ALLOWED
112306  104450                                                                  TRAP    C$MANI
112310                                  BNCOMPLETE INIT15                ;BRANCH IF ALLOWED
112310  103013                                                                  BCC     INIT15
112312                                  GMANIL INITWC,TEMP,1,NO          ;ASK OPERATOR
112312  104443                                                                  TRAP    C$GMAN
112314  000404                                                                  BR      10000$
112316  065102                                                                  .WORD   TEMP
112320  000120                                                                  .WORD   T$CODE
112322  065367                                                                  .WORD   INITWC
112324  000001                                                                  .WORD   1
112326                                                                  10000$:
112326  032737  000001  065102          BIT #1,TEMP                     ;LOOK AT RESPONSE
112334  001001                          BNE INIT15                      ;BRANCH IF YES WAS ANSWER
112336                                  DOCLN                           ;ABORT PROGRAM
112336  104444                                                                  TRAP    C$DCLN

                                ;SAVE CURRENT PARAMETERS TO FREE MEMORY SO EACH TEST CAN USE ALL OF IT

112340  013737  065016  065022  INIT15: MOV FFREE,FMEM                  ;SAVE START ADDRESS
112346  013737  065020  065024          MOV FSIZE,FMEMS                 ;SAVE SIZE

112354                          INITXX: SETPRI  #PRI00                  ; SET RUNNING PRIORITY TO ZERO
112354  012700  000000                                                          MOV     #PRI00,R0
112360  104441                                                                  TRAP    C$SPRI
112362  005037  065262                  CLR DLL                         ;ERASE DOWNLINE LOAD DATA
112366  004737  107026                  CALL CLOSEF                     ;MAKE SURE DATA FILE IS CLOSED
112372                                  READEF #EF.START
112372  012700  000040                                                          MOV     #EF.START,R0
112376  104447                                                                  TRAP    C$REFG
112400                                  BCOMPLETE INITIM
112400  103404                                                                  BCS     INITIM
112402                                  READEF #EF.RESTART
112402  012700  000037                                                          MOV     #EF.RESTART,R0
112406  104447                                                                  TRAP    C$REFG
112410                                  BNCOMPLETE KPRI
112410  103006                                                                  BCC     KPRI
112412  012701  065236          INITIM: MOV     #STIME,R1               ;AT 15 MINUTES FROM NOW
112416  012700  001604                  MOV #15.*60.,R0                 ;SET TIME FOR NEXT REPORT
112422  004737  105736                  CALL SETTO
112426                          KPRI:   EXIT INIT
112426  104432                                                                  TRAP    C$EXIT
112430  000066                                                                  .WORD   L10046-.
```

```
     1                                  ;DIFFERENT VECTORS, BR LEVELS OR BURST RATES FOR ONE CONTROLLER
     2 112432  010305                   CTABER: MOV R3,R5                       ;GET CONTROLLER ADDRESS
     3 112434                                   ERRSF 1,,ERR001
       112434  104454                                                           TRAP    C$ERSF
       112436  000001                                                           .WORD   1
       112440  000000                                                           .WORD   0
       112442  074622                                                           .WORD   ERR001
     4 112444                                   DOCLN
       112444  104444                                                           TRAP    C$DCLN
     5
     6                                  ;TWO P-TABLES FOR SAME DRIVE
     7 112446  013705  065102           MLDRER: MOV TEMP,R5                     ;GET CONTROLLER ADDRESS
     8 112452                                   ERRSF 2,,ERR002
       112452  104454                                                           TRAP    C$ERSF
       112454  000002                                                           .WORD   2
       112456  000000                                                           .WORD   0
       112460  074640                                                           .WORD   ERR002
     9 112462                                   DOCLN
       112462  104444                                                           TRAP    C$DCLN
    10
    11                                  ;MORE THAN EIGHT DRIVES SELECTED ON ONE CONTROLLER
    12
    13 112464  013705  065102           TOOMER: MOV TEMP,R5                     ;GET CONTROLLER ADDRESS
    14 112470                                   ERRSF 3,,ERR003
       112470  104454                                                           TRAP    C$ERSF
       112472  000003                                                           .WORD   3
       112474  000000                                                           .WORD   0
       112476  074656                                                           .WORD   ERR003
    15 112500                                   DOCLN
       112500  104444                                                           TRAP    C$DCLN
    16
    17                                  ;TWO UDA'S USE THE SAME VECTOR
    18
    19 112502  010305                   SAMVEC: MOV     R3,R5                   ;GET CONTROLLER ADDRESS
    20 112504                                   ERRSF   8,,ERR008
       112504  104454                                                           TRAP    C$ERSF
       112506  000010                                                           .WORD   8
       112510  000000                                                           .WORD   0
       112512  074756                                                           .WORD   ERR008
    21 112514                                   DOCLN
       112514  104444                                                           TRAP    C$DCLN
    22
    23 112516                                   ENDINIT
       112516                                                           L10046:
       112516  104411                                                           TRAP    C$INIT
```

```
                                .SBTTL  AUTODROP SECTION

                                ;++
                                ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
                                ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
                                ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
                                ; DROPPED FROM TESTING.
                                ;--

112520                                  BGNAUTO
112520                                                                  L$AUTO::

112520                                  ENDAUTO
112520                                                                  L10047:
112520  104461                                                                  TRAP    CSAUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
                                ;--

112522                                  BGNCLN
112522                                                                  L$CLEAN::

112522  004737  107026                  CALL CLOSEF                     ;CLOSE DATA FILE

112526  022737  000004  065050          CMP     #4,TNUM                 ; ARE WE DOING TEST #4?
112534  001402                          BEQ     1$                      ; IF SO, DON'T RESET BUS
112536  004737  107474                  CALL    RESET
112542                          1$:

112542                                  ENDCLN
112542                                                                  L10050:
112542  104412                                                                  TRAP    C$CLEAN

112544                                  ENDMOD
```

```
                                        .SBTTL  TEST 1: UNIBUS ADDRESSING TEST

112544                                  BGNMOD

112544                                  BGNTST
112544                                                                                  T1::
112544  012701  000001                  MOV     #1,R1                   ; INITIALIZE TEST PARAMETERS
112550  004737  076120                  CALL    TINIT
112554  013737  065026  065032          MOV     CTABS,TSTTAB            ; GET ADDRESS OF FIRST CONTROLLER TABLE
112562  013705  065032          T1NEXT: MOV     TSTTAB,R5               ; GET CONTROLLER TABLE ADDRESS
112566  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; CHECK IF UNIT AVAILABLE FOR TESTING
112574  005765  000002                  TST     C.UNIT(R5)
112600  100010                          BPL     T1NOW                   ; TEST IF AVAILABLE
112602                                  ASSUME CT.AVL EQ BIT15
112602  062737  000046  065032  T1SKIP: ADD     #C.SIZE,TSTTAB          ; MOVE TO NEXT CONTROLLER
112610  005777  152216                  TST     @TSTTAB                 ; CHECK IF ANOTHER CONTROLLER TABLE
112614  001362                          BNE     T1NEXT
112616                                  EXIT TST
112616  104432                                                                  TRAP    C$EXIT
112620  000776                                                                  .WORD   L10051-.

112622  004737  107474          T1NOW:  CALL    RESET                   ; RESET ALL UNITS
```

```
      1 112626                          BGNSUB: 1
        112626                                                          T1.1:
        112626  104402                                                          TRAP    C$BSUB
      2 112630  005037  065242                CLR NXMAD                 ;CLEAR MEMORY ERROR FLAG
      3 112634                              SETVEC #4,#NXMI,#PRI07
        112634  012746  000340                                                  MOV     #PRI07,-(SP)
        112640  012746  105716                                                  MOV     #NXMI,-(SP)
        112644  012746  000004                                                  MOV     #4,-(SP)
        112650  012746  000003                                                  MOV     #3,-(SP)
        112654  104437                                                         TRAP    C$SVEC
        112656  062706  000010                                                  ADD     #10,SP
      4 112662  011504                      MOV (R5),R4               ;GET ADDRESS OF UDAIP REGISTER
      5 112664  005714                      TST (R4)                  ;READ UDAIP
      6 112666  005764  000002              TST 2(R4)                 ;READ UDASA
      7 112672                              CLRVEC  #4                ;GIVE UP VECTOR
        112672  012700  000004                                                  MOV     #4,R0
        112676  104436                                                         TRAP    C$CVEC
      8 112700  005737  065242              TST NXMAD                 ;CHECK FLAG
      9 112704  001406                      BEQ T1GOOD
     10 112706                              ERRDF 38,,ERR038
        112706  104455                                                         TRAP    C$ERDF
        112710  000046                                                        .WORD   38
        112712  000000                                                        .WORD   0
        112714  075456                                                        .WORD   ERR038
     11 112716                              CKLOOP
        112716  104406                                                         TRAP    C$CLP1
     12 112720  000730                      BR T1SKIP                 ;END TEST NOW
     13 112722                          T1GOOD:
     14 112722                          ENDSUB
        112722                                                          L10052:
        112722  104403                                                          TRAP    C$ESUB
```

```
      1 112724                          BGNSUB; 2
        112724
        112724  104402                                                  T1.2:
                                                                                TRAP    C$BSUB
      2 112726                          DIATST:
      3                                 ;
      4                                 ;       MAKE SURE UDA PASSES INTERNAL DIAGNOSTIC
      5                                 ;       MAKE SURE UDA CAN SENSE STEP 1 AND 2
      6                                 ;
      7 112726  005014                          CLR     (R4)                    ; INIT UDA
      8 112730  012737  004000  106762         MOV     #SA.S1,UDARSD           ; STEP 1 ASSERTED?
      9 112736  004737  106624                 CALL    UDARSP                  ; WAIT FOR RESPONSE
     10 112742  103410                         BCS     1$                      ; IF FAIL, EXIT
     11 112744  012764  100000  000002         MOV     #SA.STP,2(R4)           ; SEND STEP 1
     12 112752  012737  010000  106762         MOV     #SA.S2,UDARSD           ; STEP 2 ASSERTED?
     13 112760  004737  106624                 CALL    UDARSP
     14 112764                          1$:
     15 112764                          ENDSUB
        112764                                                          L10053:
        112764  104403                                                          TRAP    C$ESUB
```

```
      1 112766                          BGNSUB: 3
        112766                                                          T1.3:
        112766  104402                                                          TRAP    C$BSUB
      2 112770                          PORTST:
      3                                 ;
      4                                 ;       TEST THE DIAGNOSTIC LOOP MODE OF ALL UDA'S ON THE SYSTEM
      5                                 ;
      6 112770  011504                          MOV     (R5),R4                 ; R4 POINTS TO UDAIP REGISTER
      7 112772                                  ASSUME C.UADR EQ 0
      8 112772  005014                          CLR     (R4)                    ; INITIALIZE THE UDA
      9 112774  012737  004000  106762          MOV     #SA.S1,UDARSD           ; LOOK FOR STEP 1
     10 113002  004737  106624                  CALL    UDARSP                  ; WAIT FOR RESPONSE
     11 113006  103444                          BCS     3$                      ; IF ERROR, BRANCH
     12 113010  016437  000002  113714          MOV     2(R4),WCHNGD            ; MOVE OLD PORT CONTENTS TO STORAGE
     13 113016  012764  140000  000002          MOV     #<SA.STP+SA.WRP>,2(R4)  ; INITIALIZE FOR PORT WRAP
     14 113024  004737  113620                  CALL    WCHNG                   ; WAIT FOR THE PORT TO CHANGE
     15 113030  001433                          BEQ     3$                      ; IF ERROR, BRANCH
     16 113032  022764  140000  000002          CMP     #<SA.STP+SA.WRP>,2(R4)  ; COMPARE WITH DATA WRITTEN
     17 113040  001017                          BNE 5$
     18 113042  012702  000001          4$:     MOV     #1,R2                   ; SET UP FOR SHIFTING '1'
     19 113046  012703  000020                  MOV     #16.,R3                 ; SET UP LOOP COUNT
     20 113052  016437  000002  113714  1$:     MOV     2(R4),WCHNGD            ; SAVE OLD PORT CONTENTS
     21 113060  010264  000002                  MOV     R2,2(R4)                ; WRITE PATTERN TO UDASA FOR LOOP
     22 113064  004737  113620                  CALL    WCHNG                   ; WAIT FOR UDASA TO CHANGE
     23 113070  001413                          BEQ     3$                      ; IF ERROR, BRANCH
     24 113072  020264  000002                  CMP     R2,2(R4)                ; COMPARE RO WITH WHAT WAS ECHOED
     25 113076  001405                          BEQ     2$                      ; IF MATCH, BRANCH
     26 113100                          5$:     ERRDF   26,,ERR026              ; REPORT ERROR
        113100  104455                                                          TRAP    C$ERDF
        113102  000032                                                          .WORD   26
        113104  000000                                                          .WORD   0
        113106  075246                                                          .WORD   ERR026
     27 113110  000403                          BR      3$                      ; BRANCH
     28 113112  006302                  2$:     ASL     R2                      ; MOVE THE SHIFTING ONE LEFT BY 1
     29 113114  005303                          DEC     R3                      ; DECREMENT COUNT
     30 113116  001355                          BNE     1$                      ; IF LOOP INCOMPLETE, BRANCH
     31 113120                          3$:
     32 113120                          ENDSUB
        113120                                                          L10054:
        113120  104403                                                          TRAP    C$ESUB
```

```
      1 113122                          BGNSUB; 4
        113122                                                          T1.4:
        113122  104402                                                          TRAP    C$BSUB
      2 113124                          INTEST:
      3                                 ;
      4                                 ;       TEST THE INTERRUPTS VECTOR AND BR LEVEL
      5
      6 113124  011504                          MOV     (R5),R4                 ; R4 POINTS TO UDAIP REGISTER
      7 113126                                  ASSUME C.UADR EQ 0
      8 113126  016503  000004                  MOV     C.VEC(R5),R3            ; GET VECTOR AND BRANCH LEVEL
      9 113132  010302                          MOV     R3,R2                   ; COPY TO R2 FOR BR LEVEL
     10 113134  042703  177000                  BIC     #^CCT.VEC,R3            ; CLEAR UNUSED VECTOR BITS
     11 113140  042702  170777                  BIC     #^CCT.BRL,R2            ; CLEAR UNUSED BRANCH LEVEL BITS
     12 113144  012701  000011                  MOV     #9.,R1                  ; SET UP TO SHIFT BR LEVEL
     13 113150  006202                  1$:     ASR     R2                      ; SHIFT BY ONE BIT
     14 113152  005301                          DEC     R1                      ; COUNT SHIFTS
     15 113154  001375                          BNE     1$                      ; IF INCOMPLETE, BRANCH
     16 113156  010237  113716                  MOV     R2,BRLEV                ; SAVE THE BRANCH LEVEL
     17 113162                                  PNTX    INTST0,(R5),R3          ; PRINT BEGINNING OF INTERRUPT MESSAGE
        113162  010346                                                                  MOV R3,-(SP)
        113164  011546                                                                  MOV (R5),-(SP)
        113166  004137  104232                                                          JSR R1,LPNTX
        113172  065746                                                                  .WORD INTST0
        113174  000004                                                                  .WORD PNT.CT
     18 113176                                  ASSUME C.UADR EQ 0
     19 113176                          SETVEC  R3,#INTSRV,#PRI00               ; SET UP INTERRUPT ROUTINE
        113176  012746  000000                                                          MOV     #PRI00,-(SP)
        113202  012746  107466                                                          MOV     #INTSRV,-(SP)
        113206  010346                                                                  MOV     R3,-(SP)
        113210  012746  000003                                                          MOV     #3,-(SP)
        113214  104437                                                                  TRAP    C$SVEC
        113216  062706  000010                                                          ADD     #10,SP
     20 113222                          SETPRI  #PRI00                          ; SET PRIORITY TO 0 TO CHECK INTERRUPTS
        113222  012700  000000                                                          MOV     #PRI00,R0
        113226  104441                                                                  TRAP    C$SPRI
     21 113230  006203                          ASR     R3                      ; DIVIDE VECTOR BY 4 FOR UDA INITIALIZATION
     22 113232  006203                          ASR     R3                      ; DIVIDE VECTOR BY 4 FOR UDA INITIALIZATION
     23 113234  052703  100200                  BIS     #<SA.STP+SA.INT>,R3     ; SET OTHER BITS FOR UDA INITIALIZATION
     24 113240  005037  065060                  CLR     INTRCV                  ; FLAG AS NO INTERRUPTS RECEIVED
     25 113244  005014                          CLR     (R4)                    ; INIT UDA
     26 113246  012737  004000  106762          MOV     #SA.S1,UDARSD           ; LOOK FOR STEP 1 COMPLETION
     27 113254  004737  106624                  CALL    UDARSP                  ; WAIT FOR COMPLETION
     28 113260  010364  000002                  MOV     R3,2(R4)                ; MOVE STEP 1 DATA TO UDA
     29 113264  012700  000012                  MOV #10.,R0                     ;SET UP TIMEOUT OF 10 SECONDS
     30 113270  010501                          MOV R5,R1
     31 113272  062701  000040                  ADD #C.TO,R1                    ;POINT TO CONTROLLER TABLE
     32 113276  004737  105736                  CALL SETTO
     33 113302  005737  065060          9$:     TST     INTRCV                  ; SEE IF INTERRUPTED
     34 113306  001016                          BNE     11$                     ; IF SO, EVERYTHING'S OK, SO BRANCH
     35 113310                                  BREAK
        113310  104422                                                                  TRAP    C$BRK
```

```
      1 113312  005737  065222                TST KW.CSR                      ;SEE IF CLOCK ON SYSTEM
      2 113316  001771                        BEQ 9$
      3 113320  023765  065234  000042        CMP KW.EL+2,C.TOH(R5)           ;SEE IF TIME ELAPSED
      4 113326  101041                        BHI 3$
      5 113330  001364                        BNE 9$
      6 113332  023765  065232  000040        CMP KW.EL,C.TO(R5)
      7 113340  103760                        BLO 9$
      8 113342  000433                        BR      3$                      ; BRANCH
      9 113344  005037  065060        11$:    CLR     INTRCV                  ; FLAG AS NO INTERRUPTS RECEIVED
     10 113350                                SETPRI  #PRI07                  ; SET PRIORITY AS HIGHEST PRIORITY
        113350  012700  000340                                                          MOV     #PRI07,R0
        113354  104441                                                                  TRAP    C$SPRI
     11 113356  005064  000002                CLR     2(R4)                   ; WRITE SECOND STEP TO UDA
     12 113362  012702  000144                MOV     #100.,R2                ; SET UP DELAY SO WE KNOW WE'RE INTERRUPTED
     13 113366  005302                12$:    DEC     R2                      ; DECREMENT COUNT
     14 113370  001376                        BNE     12$                     ; IF INCOMPLETE, BRANCH
     15 113372  012701  000007                MOV     #7.,R1                  ; R1 IS PROCESS PRIORITY LEVEL
     16 113376                        2$:     PUSH    R1                      ; SAVE PRIORITY
        113376  010146                                                                  MOV R1,-(SP)
     17 113400  012702  000005                MOV     #5.,R2                  ; SET UP FOR SHIFTING PRIORITY
     18 113404  006301                10$:    ASL     R1                      ; SHIFT PRIORITY
     19 113406  005302                        DEC     R2                      ; DECREMENT SHIFT COUNT
     20 113410  001375                        BNE     10$                     ; IF INCOMPLETE, BRANCH
     21 113412                                SETPRI  R1                      ; SET RUNNING PRIORITY TO R1
        113412  010100                                                                  MOV     R1,R0
        113414  104441                                                                  TRAP    C$SPRI
     22 113416                                POP     R1                      ; RESTORE R1
        113416  012601                                                                  MOV (SP)+,R1
     23 113420  005737  065060                TST     INTRCV                  ; SEE IF INTERRUPT RECEIVED
     24 113424  001007                        BNE     4$                      ; IF SO, BRANCH
     25 113426  005301                        DEC     R1                      ; DECREMENT PRIORITY LEVEL
     26 113430  100362                        BPL     2$                      ; IF ALL LEVELS UNTESTED, BRANCH
     27 113432                        3$:     ERRDF   28,,ERR028              ; REPORT NO INTERRUPTS ERROR
        113432  104455                                                                  TRAP    C$ERDF
        113434  000034                                                                  .WORD   28
        113436  000000                                                                  .WORD   0
        113440  075304                                                                  .WORD   ERR028
     28 113442  000420                        BR      6$                      ; BRANCH
```

```
      1 113444                          4$:     SETPRI  #PRI00          ; SET RUNNING PRIORITY TO 0
        113444  012700  000000                                                  MOV     #PRI00,R0
        113450  104441                                                          TRAP    C$SPRI
      2 113452  005201                          INC     R1              ; SO PRIORITY = BR LEVEL
      3 113454  023701  113716                  CMP     BRLEV,R1        ; SEE IF BR LEVEL MATCHES PRIORITY
      4 113460  001405                          BEQ     5$              ; IF SO, BRANCH
      5 113462                                  ERRDF   29,,ERR029      ; REPORT ERROR
        113462  104455                                                          TRAP    C$ERDF
        113464  000035                                                          .WORD   29
        113466  000000                                                          .WORD   0
        113470  075316                                                          .WORD   ERR029
      6 113472  000404                          BR      6$              ; BRANCH
      7 113474                          5$:     PNTX    INTST1          ; PRINT TESTING COMPLETED
        113474  004137  104232                                                  JSR R1,LPNTX
        113500  066043                                                          .WORD INTST1
        113502  000000                                                          .WORD PNT.CT
      8 113504  016503  000004          6$:     MOV     C.VEC(R5),R3    ; GET VECTOR ADDRESS
      9 113510  042703  177000                  BIC     #^CCT.VEC,R3    ; CLEAR UNUSED BITS
     10 113514                                  CLRVEC  R3              ; CLEAR VECTOR
        113514  010300                                                          MOV     R3,R0
        113516  104436                                                          TRAP    C$CVEC
     11 113520                          ENDSUB
        113520                                                          L10055:
        113520  104403                                                          TRAP    C$ESUB
```

```
      1 113522                          BGNSUB; 5
        113522                                                          T1.5:
        113522  104402                                                          TRAP    C$BSUB
      2 113524  005004                          CLR     R4                ; INITIALIZE UDA WITH SMALLEST
      3 113526  004737  106020                  CALL    UDAINT            ; RING BUFFER AND INTERRUPTS DISABLED
      4 113532                          ENDSUB
        113532                                                          L10056:
        113532  104403                                                          TRAP    C$ESUB
```

```
      1 113534                          BGNSUB; 6
        113534                                                                  T1.6:
        113534  104402                                                                  TRAP    C$BSUB
      2 113536  012704  126400                MOV     #<SA.STP+<5*SA.MS1>+<5*SA.CM1>>,R4      ;INITIALIZE UDA WITH RING BUFFER
      3 113542  004737  106020                CALL    UDAINT                          ; LARGE ENOUGH TO COVER NORMAL HOST COMM AREA
      4                                                                             ; PACKET AND BUFFER SPACE (A 5 IN MES
      5                                                                             ; LENGTH AND A 5 IN CMD LENGTH)
      6 113546                          ENDSUB
        113546                                                                  L10057:
        113546  104403                                                                 TRAP    C$ESUB
```

```
      1 113550                          BGNSUB: 7
        113550
        113550  104402                                                  T1.7:
                                                                                TRAP    C$BSUB
      2 113552                                  PUSH    FFREE         ; SAVE FREE MEMORY PARAMETERS
        113552  013746  065016                                                  MOV FFREE,-(SP)
      3 113556                                  PUSH    FSIZE
        113556  013746  065020                                                  MOV FSIZE,-(SP)
      4 113562  012701  000001                  MOV     #1,R1         ; RUN DM PROGRAM IN
      5 113566  004737  077022                  CALL    RUNDM         ; ONE CONTROLLER ONLY
      6 113572  001402                          BEQ 1$
      7 113574  004737  077120                  CALL RESPDM
      8 113600                          1$:     POP     FSIZE
        113600  012637  065020                                                  MOV (SP)+,FSIZE
      9 113604                                  POP     FFREE
        113604  012637  065016                                                  MOV (SP)+,FFREE
     10 113610                          ENDSUB
        113610                                                          L10060:
        113610  104403                                                         TRAP    C$ESUB
     11
     12 113612  000137  112602                  JMP T1SKIP
     13
     14 113616                          ENDTST
        113616                                                          L10051:
        113616  104401                                                         TRAP    C$ETST
```

TEST 1: UNIBUS ADDRESSING TEST

```
                                        ;WCHNG
                                        ;
                                        ;       WAIT UNTIL UDASA CHANGES FROM WHAT IS IN WCHNGD
                                        
113620  012700  000012          WCHNG:  MOV #10.,R0                      ;SET TIMEOUT FOR 10 SECONDS
113624  010501                          MOV R5,R1                       ;POINT TO CONTROLLER TABLE
113626  062701  000040                  ADD #C.TO,R1
113632  004737  105736                  CALL SETTO
113636  026437  000002  113714  1$:     CMP 2(R4),WCHNGD                ;SEE IF CHANGED
113644  001022                          BNE 2$
113646                                  BREAK
113646  104422                                                          TRAP    C$BRK
113650  005737  065222                  TST KW.CSR                      ;SEE IF CLOCK ON SYSTEM
113654  001770                          BEQ 1$
113656  023765  065234  000042          CMP KW.EL+2,C.TOH(R5)           ;CHECK IF TIME OUT OCCURRED
113664  101005                          BHI 3$
113666  001363                          BNE 1$
113670  023765  065232  000040          CMP KW.EL,C.TO(R5)
113676  103757                          BLO 1$
113700                          3$:     ERRDF   27,,ERR027              ; REPORT ERROR
113700  104455                                                          TRAP    C$ERDF
113702  000033                                                          .WORD   27
113704  000000                                                          .WORD   0
113706  075266                                                          .WORD   ERR027
113710  000264                          SEZ                             ; FLAG AS ERROR
113712  000207                  2$:     RETURN                          ; RETURN TO CALLING PROGRAM

113714                          WCHNGD: .BLKW   1                       ; OLD PORT CONTENTS
113716                          BRLEV:  .BLKW   1                       ; WORD FOR BRANCH LEVEL STORAGE
```

```
                                .SBTTL  TEST 2: DISK RESIDENT DIAGNOSTIC TEST

                                        BGNTST
                                                                                T2::

                                        MOV #2,R1               ;INIT TEST PARAMETERS
                                        CALL TINIT

                                        MOV CTABS,TSTTAB        ;GET POINTER TO FIRST CONTROLLER TABLE

                                T2NEXT: CALL RESET              ;RESET ALL UNITS
                                        PUSH FFREE              ;SAVE FREE MEMORY PARAMETERS
                                                                                MOV FFREE,-(SP)
                                        PUSH FSIZE
                                                                                MOV FSIZE,-(SP)
                                        MOV #1,R1               ;RUN DM PROGRAM IN
                                        CALL RUNDM              ; ONE CONTROLLER ONLY
                                        BEQ 1$
                                        CALL RESPDM
                                1$:     POP FSIZE
                                                                                MOV (SP)+,FSIZE
                                        POP FFREE
                                                                                MOV (SP)+,FFREE

                                        ADD #C.SIZE,TSTTAB      ;MOVE TO NEXT CONTROLLER
                                        TST @TSTTAB             ;CHECK IF ANY MORE CONTROLLER TABLES
                                        BNE T2NEXT

                                        ENDTST
                                                                        L10061:
                                                                                TRAP    C$ETST
```

```
                                .SBTTL  TEST 3: DISK FUNCTION TEST

                                        BGNTST
                                                                        T3::
                                        MOV #3,R1                       ;INITIALIZE TEST PARAMETERS
                                        CALL TINIT

                                        MOV CTABS,TSTTAB                ;GET FIRST TABLE ADDRESS
                                        MOV CTRLRS,R1                   ;RUN DM PROGRAM ON ALL CONTROLLERS
                                        CALL RUNDM                      ;  AT ONCE
                                        BEQ 1$
                                        CALL RESPDM
                                1$:
                                        ENDTST
                                                                        L10062:
                                                                                TRAP    C$ETST
```

```
                                .SBTTL  TEST 4: DISK EXERCISER

                                        BGNTST
                                                                        T4::
                                        CMP #4,TNUM                     ;CHECK IF TEST 4 WAS IN PROGRESS
                                        BNE T4STRT                      ;BRANCH IF NOT
                                        CMP #ICONT,IFLAGS               ;CHECK IF HERE BY CONTINUE COMMAND
                                        BNE T4STRT                      ;BRANCH IF NOT
                                        CLR IFLAGS                      ;CLEAR FLAGS FOR NEXT TIME HERE
                                        MOV LBUFS,R4                    ;GET LOG BUFFER POINTER
                                        BEQ LOGCHK                      ;  IF ZERO, NONE EXISTS
                                        PNTF LOGM1                      ;INTRODUCE ERROR LOG
                                                                                JSR R1,LPNTF
                                                                                .WORD LOGM1
                                                                                .WORD PNT.CT
                                        CLR LBUFS                       ;CLEAR START ADDRESS TO ERASE BUFFER
LOGOUT:                                 MOV (R4)+,R5                    ;GET CONTROLLER TABLE ADDRESS
                                        CALL PNTERR                     ;PRINT ERROR REPORT
                                        ADD #<HC.BSZ-2>,R4              ;BUMP POINTER TO NEXT ENTRY
                                        CMP R4,LBUFN                    ;CHECK IF AT END
                                        BLO LOGOUT                      ;PRINT ALL ENTRIES
                                        PNTF LOGM2
                                                                                JSR R1,LPNTF
                                                                                .WORD LOGM2
                                                                                .WORD PNT.CT
                                        BR T4CON

LOGCHK:                 BIT #SM.LOG,SFPTBL+SO.BIT       ;CHECK IF LOG ENABLED
                        BEQ T4CON
                        PNTF LOGM3                      ;REPORT LOG EMPTY
                                                                JSR R1,LPNTF
                                                                .WORD LOGM3
                                                                .WORD PNT.CT
T4CON:                  TST URNING                      ;CHECK IF ANY CONTROLLERS STILL RUNNING
                        BEQ T4STRT                      ;RESTART IF NOT
                        CALL RESPDM                     ;CONTINUE BY RESPONDING TO REQUESTS
                        JMP T4WAIT                      ;END OF TEST WHEN DONE
```

```
                                        ;START TEST 4
                                        T4STRT:
114212  012701  000004                          MOV #4,R1                       ;INITIALIZE TEST PARAMETERS
114216  004737  076120                          CALL TINIT
                                        
114222  032737  000014  065044                  BIT #ISTRT+IREST,IFLAGS         ;HERE FROM OPERATOR COMMAND?
114230  001521                                  BEQ T4RUN                       ;RUN WITH PREVIOUS PARAMETERS IF NEW PASS
114232  032737  000200  065000                  BIT #SM.MAN,SFPTBL+SO.BIT       ;MANUAL INTERVENTION MODE?
114240  001463                                  BEQ T4DEF                       ;IF NOT, SET UP DEFAULT PARAMETERS
114242                                          MANUAL                          ;MANUAL INTERVENTION ALLOWED?
114242  104450                                                                          TRAP    C$MANI
114244                                          BNCOMPLETE T4DEFW               ;IF NOT, GIVE WARNING
114244  103055                                                                          BCC     T4DEFW
114246  012701  000007                          MOV     #7,R1                   ; R1 = T4QUEST FILE NUMBER
114252  020137  065046                          CMP     R1,FNUM                 ; IS IT ALREADY LOADED?
114256  001406                                  BEQ     1$                      ; IF SO, BRANCH
114260  005005                                  CLR     R5                      ; ELSE R5 = ADJUSTED ADDRESS
114262  004737  107044                          CALL    RDREC                   ; READ IN FILE
114266  103002                                  BCC     1$                      ; IF OK, BRANCH
114270  000137  076242                          JMP     TINITE                  ; ELSE, ERROR
                                        
                                        ;INPUT PARAMETERS
                                        
114274  005037  065056          1$:             CLR UCNT                        ;CLEAR COUNT OF UNITS USING PATTERN 16
114300  013705  065026                          MOV CTABS,R5                    ;GET FIRST CONTROLLER TABLE
114304  012702  000010          T4PRM1:         MOV #8.,R2                      ;GET COUNT OF DRIVE TABLES
114310  010504                                  MOV R5,R4                       ;GET FIRST DRIVE TABLE POINTER
114312  062704  000020                          ADD #C.DRO,R4
114316  012403                  T4PRM2:         MOV (R4)+,R3                    ;GET DRIVE TABLE ADDRESS
114320  001416                                  BEQ T4PRM4                      ;GO TO NEXT CONTROLLER IF NONE
114322  032763  100000  000002                  BIT #DT.AVL,D.UNIT(R3)          ;SEE IF TO BE TESTED
114330  001010                                  BNE T4PRM3
114332  004737  002122                          CALL STORAG                     ;ASK QUESTIONS
114336  022763  000020  000006                  CMP #16.,D.PAT(R3)
114344  001002                                  BNE T4PRM3
114346  005237  065056                          INC UCNT
114352  005302                  T4PRM3:         DEC R2                          ;COUNT DRIVE TABLES
114354  001360                                  BNE T4PRM2                      ;GO LOOK AT NEXT
114356  062705  000046          T4PRM4:         ADD #C.SIZE,R5                  ;GO TO NEXT CONTROLLER
114362  005715                                  TST (R5)                        ;  IF THERE IS ONE
114364  001347                                  BNE T4PRM1
114366  012701  065160                          MOV     #PAT16C,R1              ; R1 -> PAT16C FOR INPUT
114372  004737  002124                          CALL    STORAG+2                ; ASK LAST QUESTIONS
114376                                          ASSUME PAT16W EQ PAT16C+2
114376  000436                                  BR      T4RUN
```

```
      1                                 ;NOW GET DATA PATTERN 16 IF SELECTED BY ANY DRIVE
      2                                 ;
      3                                 ;GIVE WARNING MANUAL INTERVENTION NOT ALLOWED
      4                                 ;
      5 114400                          T4DEFW: PNTF T4WARN
        114400  004137  104212                                          JSR R1,LPNTF
        114404  066211                                                  .WORD T4WARN
        114406  000000                                                  .WORD PNT.CT
```

```
                                        ;SET UP DEFAULT PARAMETERS
                                        
114410  013705  065026          T4DEF:  MOV CTABS,R5                    ;GET FIRST CONTROLLER TABLE
114414  012702  000010          T4DEFA: MOV #8.,R2                      ;GET COUNT OF DRIVE TABLES
114420  010504                          MOV R5,R4                       ;GET FIRST DRIVE TABLE POINTER
114422  062704  000020                  ADD #C.DRO,R4
114426  012403                  T4DEFB: MOV (R4)+,R3                    ;GET DRIVE TABLE ADDRESS
114430  001415                          BEQ T4DEFE                      ;GO TO NEXT CONTROLLER IF NONE
114432  062703  000004                  ADD #D.PRM,R3
114436                                  AND D.DCY,(R3)                  ;INITIALIZE ALL PARAMETER BITS
114436  042713  157777                                                                  BIC #^C<D.DCY>,(R3)
114442  052723  011012                  BIS #DDEF,(R3)+
114446  012700  000067                  MOV #55.,R0
114452  005023                  T4DEFC: CLR (R3)+
114454  005300                          DEC R0
114456  001375                          BNE T4DEFC
114460  005302                  T4DEFD: DEC R2                          ;COUNT DRIVE TABLES
114462  001361                          BNE T4DEFB                      ;GO LOOK AT NEXT
114464  062705  000046          T4DEFE: ADD #C.SIZE,R5                  ;GO TO NEXT CONTROLLER
114470  005715                          TST (R5)                        ;  IF THERE IS ONE
114472  001350                          BNE T4DEFA
                                        
                                        ;START TEST 4
                                        
114474  006137  065044          T4RUN:  ROL IFLAGS                      ;CLEAR FLAGS FOR NEXT TIME HERE
114500                                  AND ISTRTH,IFLAGS               ;HOLD START FOR T4UPRM REQUEST
114500  042737  177757  065044                                                        BIC #^C<ISTRTH>,IFLAGS
114506  013737  065026  065032          MOV CTABS,TSTTAB                ;GET FIRST TABLE ADDRESS
114514  013701  065030                  MOV CTRLRS,R1                   ;RUN DM PROGRAM ON ALL CONTROLLERS
114520  004737  076254                  CALL RNT4DM                     ;  AT ONCE
114524  001405                          BEQ T4WAIT
114526  013737  065026  065032          MOV     CTABS,TSTTAB            ; MAKE SURE TSTTAB HAS CONTROLLER INFO
114534  004737  077120                  CALL RESPDM
114540  032737  001000  065000  T4WAIT: BIT #SM.LOG,SFPTBL+SO.BIT        ;CHECK IF LOG IS ENABLED
114546  001402                          BEQ T4EXIT                      ;EXIT IF NOT
114550                                  BREAK
114550  104422                                                          TRAP    C$BRK
114552  000772                          BR T4WAIT                       ;WAIT TILL STOPPED BY CONTROL C
                                        
114554                          T4EXIT: DORPT                           ;PRINT STATISTICS
114554  104424                                                          TRAP    C$DRPT
114556                                  EXIT TST
114556  104432                                                          TRAP    C$EXIT
114560  000002                                                          .WORD   L10063-.
                                        
114562                                  ENDTST
114562                                                          L10063:
114562  104401                                                          TRAP    C$ETST
114564                                  ENDMOD
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION

114564                                  BGNMOD

                                ;++
                                ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

114564                                  BGNHRD
114564  000032                                                          .WORD L10064-L$HARD/2
114566                                                          L$HARD::

                                ;FORMAT OF HARDWARE P-TABLE IS AS FOLLOWS:

114566                          TABLE                                   ;START A TEBLE DEFINITION

114566                          ITEM HO.UBA     2                       ; UNIBUS ADDRESS
114566                          ITEM HO.VEC     2                       ; UDA VECTOR
114566                          ITEM HO.BRL     2                       ; BR LEVEL
114566                          ITEM HO.BST     2                       ; BURST RATE
114566                          ITEM HO.LDR     2                       ; DRIVE NUMBER
114566                          ITEM HO.PRM     2                       ; PROGRAM PARAMETERS
        020000                          HM.CYL  ==      BIT13           ; TEST CUSTOMER DATA AREA
114566                          END
```

HARDWARE PARAMETER CODING SECTION

```
     1 114566                                GPRMA   H.UBA,HO.UBA,O,160000,177774,YES        ;BUS ADDRESS
       114566  000031                                                                          .WORD   T$CODE
       114570  114652                                                                          .WORD   H.UBA
       114572  160000                                                                          .WORD   T$LOLIM
       114574  177774                                                                          .WORD   T$HILIM
     2 114576                                GPRMA   H.VEC,HO.VEC,O,4,774,YES         ; VECTOR
       114576  001031                                                                          .WORD   T$CODE
       114600  114700                                                                          .WORD   H.VEC
       114602  000004                                                                          .WORD   T$LOLIM
       114604  000774                                                                          .WORD   T$HILIM
     3 114606                                GPRMD   H.BRL,HO.BRL,D,-1,4.,7.,YES     ; BR LEVEL
       114606  002052                                                                          .WORD   T$CODE
       114610  114707                                                                          .WORD   H.BRL
       114612  177777                                                                          .WORD   -1
       114614  000004                                                                          .WORD   T$LOLIM
       114616  000007                                                                          .WORD   T$HILIM
     4 114620                                GPRMD   H.BST,HO.BST,D,-1,0.,63.,YES     ; BURST RATE
       114620  003052                                                                          .WORD   T$CODE
       114622  114720                                                                          .WORD   H.BST
       114624  177777                                                                          .WORD   -1
       114626  000000                                                                          .WORD   T$LOLIM
       114630  000077                                                                          .WORD   T$HILIM
     5 114632                                GPRMD   H.LDR,HO.LDR,D,-1,0.,255.,YES   ; DRIVE SELECT NUMBER
       114632  004052                                                                          .WORD   T$CODE
       114634  114742                                                                          .WORD   H.LDR
       114636  177777                                                                          .WORD   -1
       114640  000000                                                                          .WORD   T$LOLIM
       114642  000377                                                                          .WORD   T$HILIM
     7 114644                                GPRML   H.CST,HO.PRM,HM.CYL,YES ; USE CUSTOMER DATA AREA
       114644  005130                                                                          .WORD   T$CODE
       114646  114757                                                                          .WORD   H.CST
       114650  020000                                                                          .WORD   HM.CYL
     9 114652                                ENDHRD
                                                                                               .EVEN
       114652                                                                          L10064:  .EVEN
    10
    11 114652     125     116     111  H.LBA:  .ASCIZ  \UNIBUS ADDRESS OF UDA\
    12 114700     126     105     103  H.VEC:  .ASCIZ  \VECTOR\
    13 114707     102     122     040  H.BRL:  .ASCIZ  \BR LEVEL\
    14 114720     125     116     111  H.BST:  .ASCIZ  \UNIBUS BURST RATE\
    15 114742     104     122     111  H.LDR:  .ASCIZ  \DRIVE NUMBER\
    17 114757     105     130     105  H.CST:  .ASCIZ  \EXERCISE ON CUSTOMER DATA AREA IN TEST 4\
    19                                        .EVEN
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

115030                                  BGNSFT
115030  000030                                                          .WORD L10065-L$SOFT/2
115032                                                          L$SOFT::

                                ;FORMAT OF SOFTWARE P-TABLE IS AS FOLLOWS:

115032                          TABLE                                   ;START A TABLE DEFINITION

115032                          ITEM SO.EL      2                       ;ERROR LIMIT
115032                          ITEM SO.XL      2                       ;DATA TRANSFER LIMIT (MEGABITS)
115032                          ITEM SO.BIT     2                       ;SINGLE BIT ANSWERS
        000200                          SM.MAN==BIT07                   ;  MANUAL INTERVENTION MODE
        000400                          SM.SSF==BIT08                   ;  SUPPRESS SOFT ERRORS
        001000                          SM.LOG==BIT09                   ;  ERROR LOG ENABLED
        040000                          SM.IW== BIT14                   ;  INITIAL WRITE

115032                          END
```

```
 115032                                          GPRML S.MAN,SO.BIT,SM.MAN,YES   ;MANUAL INTERVENTION MODE
 115032  002130                                                                                  .WORD   T$CODE
 115034  115112                                                                                  .WORD   S.MAN
 115036  000200                                                                                  .WORD   SM.MAN
 115040                                          DISPLAY S.MES                   ;MESSAGE ON NEXT QUESTIONS
 115040  000003                                                                                  .WORD   T$CODE
 115042  115177                                                                                  .WORD   S.MES
 115044                                          GPRMD S.EL,SO.EL,D,-1,1.,-1.,YES        ;ERROR LIMIT
 115044  000052                                                                                  .WORD   T$CODE
 115046  115262                                                                                  .WORD   S.EL
 115050  177777                                                                                  .WORD   -1
 115052  000001                                                                                  .WORD   T$LOLIM
 115054  177777                                                                                  .WORD   T$HILIM
 115056                                          GPRMD S.XL,SO.XL,D,-1,0.,-1.,YES        ;TRANSFER LIMIT
 115056  001052                                                                                  .WORD   T$CODE
 115060  115276                                                                                  .WORD   S.XL
 115062  177777                                                                                  .WORD   -1
 115064  000000                                                                                  .WORD   T$LOLIM
 115066  177777                                                                                  .WORD   T$HILIM
 115070                                          GPRML S.SSF,SO.BIT,SM.SSF,YES   ;SUPPRESS SOFT ERRORS
 115070  002130                                                                                  .WORD   T$CODE
 115072  115360                                                                                  .WORD   S.SSF
 115074  000400                                                                                  .WORD   SM.SSF
 115076                                          GPRML S.IW,SO.BIT,SM.IW,YES     ;INITIAL WRITE
 115076  002130                                                                                  .WORD   T$CODE
 115100  115416                                                                                  .WORD   S.IW
 115102  040000                                                                                  .WORD   SM.IW
 115104                                          GPRML S.LOG,SO.BIT,SM.LOG,YES   ;ERROR LOG
 115104  002130                                                                                  .WORD   T$CODE
 115106  115450                                                                                  .WORD   S.LOG
 115110  001000                                                                                  .WORD   SM.LOG
 115112                                          ENDSFT
                                                                                                 .EVEN
 115112                                                                                L10065:

 115112      105       116       124   S.MAN:    .ASCIZ\ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS\
 115177      122       105       115   S.MES:    .ASCIZ\REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY\
 115261      000                                 .BYTE 0
 115262      105       122       122   S.EL:     .ASCIZ\ERROR LIMIT\
 115276      122       105       101   S.XL:     .ASCIZ\READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT\
 115360      123       125       120   S.SSF:    .ASCIZ\SUPPRESS PRINTING SOFT ERRORS\
 115416      104       117       040   S.IW:     .ASCIZ\DO INITIAL WRITE ON START\
 115450      105       116       101   S.LOG:    .ASCIZ\ENABLE ERROR LOG\
                                                 .EVEN
```

PATCH AREA

```
                                        .SBTTL  PATCH AREA
                                
115472                          $PATCH::
        000050                          .REPT 40.
                                        .WORD 0
                                        .ENDR
```

PATCH AREA

```
      8
      9 115612                              LASTAD
                                                                        .EVEN
        115612  115636                                                .WORD T$FREE
        115614  000010                                                .WORD T$SIZE
        115616                        L$LAST::
     10
     11 115616                              ENDMOD
```

PATCH AREA

```
      1 115616                          BGNSETUP        1
      2
      3 115616                          BGNPTAB
        115616  000000                                                .WORD   0
        115620  000006                                                .WORD   L10070-./2-1
        115622                                                  L10066:
      4
      5 115622  172150                  .WORD   172150          ; UNIBUS ADDRESS
      6 115624  000154                  .WORD   154             ; VECTOR ADDRESS
      7 115626  000005                  .WORD   5.              ; BR LEVEL
      8 115630  000077                  .WORD   63.             ; UNIBUS BURST RATE
      9 115632  000000                  .WORD   0.              ; LOGICAL DRIVE NUMBER
     10 115634  000000                  .WORD   0               ; CUSTOMER DATA AREA
     11
     12 115636                          ENDPTAB
        115636                                                  L10070:
     13
     14 115636                          ENDSETUP
     15
     16
     17
     18
     19
     20
     21
     22         000001          .END
ERRORS DETECTED:  1

VIRTUAL MEMORY USED:  29952 WORDS  ( 117 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  70 PAGES
A:ZUDCC0,B:ZUDCC0/C=[20,0]SYC34R.MLB/P:1,ZUDCC0.DOC,ZUDCC0.MAC
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

```
BIT9    121-10#  133-19
BLDC0   219-22   219-24#
BLDC1   219-26#  219-28
BLDCMD  159-15   159-34   213-7    215-17   218-17   219-15#
BOE     121-10#
BRLEV   270-16*  272-3    276-25#
C$AU    115-12#
C$AUTO  115-12#  264-12
C$BRK   115-12#  156-43   222-26   231-13   234-27   239-31   245-12   270-35   276-11   282-45
C$BSEG  115-12#
C$BSUB  115-12#  267-1    268-1    269-1    270-1    273-1    274-1    275-1
C$CEFG  115-12#
C$CLCK  115-12#  255-13   255-15
C$CLEA  115-12#  265-21
C$CLOS  115-12#  238-13
C$CLP1  115-12#  267-11
C$CVEC  115-12#  231-27   267-7    272-10
C$DCLN  115-12#  147-8    150-39   243-7    245-30   254-30   262-9    263-4    263-9    263-15   263-21
C$DODU  115-12#
C$DRPT  115-12#  151-32   157-10   245-29   253-12   282-48
C$DU    115-12#
C$EDIT  115-12#  115-42
C$ERDF  115-12#  151-31   156-22   156-40   158-21   159-11   202-24   216-3    222-34   223-5    229-18   229-40   231-30   233-50
        234-36   235-5    245-27   267-10   269-26   271-27   272-5    276-19
C$ERHR  115-12#
C$ERRO  115-12#  211-41
C$ERSF  115-12#  147-7    150-38   153-1    243-6    254-29   263-3    263-8    263-14   263-20
C$ERSO  115-12#
C$ESCA  115-12#
C$ESEG  115-12#
C$ESUB  115-12#  267-14   268-15   269-32   272-11   273-4    274-6    275-10
C$ETST  115-12#  275-14   277-24   278-14   282-51
C$EXIT  115-12#  262-31   266-18   282-49
C$GETB  115-12#  239-33   239-38   240-7    241-17   241-24   242-1    242-12   242-15
C$GETW  115-12#
C$GMAN  115-12#  170-3    262-6
C$GPHR  115-12#  254-21   256-15   259-5
C$GPLO  115-12#
C$GPRI  115-12#
C$INIT  115-12#  263-23
C$INLP  115-12#
C$MANI  115-12#  167-33   262-4    280-15
C$MEM   115-12#  255-2
C$MSG   115-12#  144-16   144-20   144-24   144-28   144-33   144-37   144-41   144-46   144-50   144-60   144-64   144-75   144-81
        144-95   144-99   144-103  144-108  144-112  144-116  144-120  144-125  144-129  144-134  144-138  144-142  144-147  144-151
        144-155  144-159  145-22
C$OPEN  115-12#  239-28
C$PNTB  115-12#  209-15
C$PNTF  115-12#  209-13   261-36
C$PNTS  115-12#  209-19   249-22   249-24
C$PNTX  115-12#  144-80   209-17
C$QIO   115-12#
C$RDBU  115-12#
C$REFG  115-12#  253-1    253-5    253-9    253-13   262-20   262-22
C$RESE  115-12#  115-12#
C$REVI  115-12#  115-42
```

CROSS REFERENCE TABLE (CREF V04.00)

```
C$RFLA  115-12#  180-49   182-37   184-44
C$RPT   115-12#  250-43
C$SEFG  115-12#
C$SPRI  115-12#  262-17   270-20   271-10   271-21   272-1
C$SVEC  115-12#  212-37   218-9    231-24   245-11   255-24   267-3    270-19
C$TPRI  115-12#
C.BST   132-19#  230-12   258-9
C.DRO   132-35#  181-19   202-15   248-21   254-7    260-11   261-30   280-33   282-6
C.DR1   132-36#
C.DR2   132-37#
C.DR3   132-38#
C.DR4   132-39#
C.DR5   132-40#
C.DR6   132-41#
C.DR7   132-42#
C.FLG   132-22#  151-29   151-36   152-15   152-27*  152-31   152-50   152-52*  155-26*  156-9    156-12   156-14   157-21*  158-6
        158-26   159-29*  159-30   159-37*  159-40*  180-53*  182-59*  186-18*  213-19*  213-20*  215-27*  217-11   217-15   220-27*
        222-17   223-1*   233-41*  233-43*
C.JAD   132-21#
C.JSR   132-20#  212-36   218-8
C.REF   132-45#  158-19   220-16*  220-17
C.RING  132-34#  144-175  149-17*  156-8    219-16   220-15
C.SIZE  132-47#  151-38   152-18   152-40   152-57   155-35   157-14   250-3    254-14   254-26   256-32   257-12   259-13   261-39
        266-15   277-20   280-48   282-18
C.TO    132-43#  156-38   159-45   222-14   222-32   234-22   234-33   270-31   271-6    276-7    276-17
C.TOH   132-44#  156-35   222-29   234-30   271-3    276-14
C.UADR  132-12#  145-9    152-17*  152-26   231-22   256-24   269-7    270-7    270-18
C.UNIT  132-13#  151-23   151-33*  152-13   152-28   152-29   152-48   152-53   155-28   155-29   156-11   211-16   248-11   254-5*
        254-34*  266-11   266-12
C.VEC   132-16#  212-33   218-5    256-27   258-7    270-8    272-8
CALR1   191-3    193-4#   195-10
CALR2   191-4    194-4#
CALR3   191-5    195-9#
CALR4   191-6    196-5#
CALR5   191-7    197-5#
CALR6   191-8    198-4#
CALR7   191-9    199-4#
CALR8   191-10   200-4#
CALR9   191-11   201-3#
CF.576  128-39#
CF.ATN  128-34#
CF.MSC  128-35#
CF.OTH  128-36#
CF.SHD  128-38#
CF.THS  128-37#
CLOG    228-26   228-34   236-10#
CLOGLP  236-13#  236-15
CLOSEF  237-18   238-11#  243-1    262-19   265-10
CLRBFL  221-22#  221-24
CLRBUF  159-17   159-36   221-16#
CON.A   189-12#  192-21
CON.A1  189-13#  189-15   194-19
CON.A2  189-17   189-19#
CON.D   189-23#  192-21
CON.H   189-29#  192-21
CON.N   190-9#   192-21
```

CROSS REFERENCE TABLE (CREF V04.00)

```
CON.N1  190-10#  190-12
CON.O   190-3#   192-21
CON.QU  189-3#   189-7    192-21
CON.QX  189-5    189-8#
CON.R   190-17#  192-21
CON.R1  190-19   190-23#
CON.S   190-29#  192-21
CON.S1  190-30#  190-32
CPNT    145-11   169-3    169-8    169-11   189-6    189-13   190-10   190-30   194-12   194-20   196-7    197-7    205-13   205-22
        207-4    207-12   207-28   209-6#   246-27   246-32   246-35   247-15
CPNTX   209-14   209-16   209-18   209-20#
CR      121-12#  145-11   169-11   190-10   194-12   194-20   196-7    197-7    209-9    247-15
CT.AVL  132-15#  155-31   248-14   254-5    254-34   266-14
CT.BRL  132-18#  270-11
CT.CMD  132-24#  156-14   213-19   215-27   220-27
CT.MSG  132-25#  156-12   157-21   159-29   180-53   182-59   186-18   222-17   223-1    226-19
CT.REQ  132-27#  158-6    158-26   159-30   159-37   159-40   213-19
CT.RN   132-23#  156-9    157-21   213-20
CT.USO  132-30#  151-29   151-36   152-15   152-27   152-31   152-50   152-52   155-26   217-11   217-15   233-41   233-43
CT.UNT  132-14#
CT.VEC  132-17#  212-34   218-6    256-28   270-10   272-9
CTABER  258-11   263-2#
CTABS   136-17#  248-9    254-3    254-23   256-3*   256-4*   256-20   259-10   261-28   266-9    277-8    278-8    280-30   282-3
        282-32   282-37
CTRLRS  136-18#  256-5*   257-28*  278-9    282-33
D.BB    133-35#  176-25
D.BB01  133-36#
D.BB02  133-37#
D.BB03  133-38#
D.BB04  133-39#
D.BB05  133-40#
D.BB06  133-41#
D.BB07  133-42#
D.BB08  133-43#
D.BB09  133-44#
D.BB10  133-45#
D.BB11  133-46#
D.BB12  133-47#
D.BB13  133-48#
D.BB14  133-49#
D.BB15  133-50#  177-18
D.BB16  133-51#
D.BCYL  134-10#  174-44   175-7
D.BE    133-22#  174-22   175-3    175-15
D.BEC   134-1#   174-29   174-40   174-50
D.BGN1  134-2#
D.BGN2  134-4#
D.BGN3  134-6#
D.BGN4  134-8#
D.CYL   133-20#  174-35
D.DC    133-26#  133-31
D.DCA   133-27#
D.DCY   133-15#  261-34   282-10
D.DRV   133-9#   249-23
D.ECC   133-16#  133-31
D.ECCC  134-17#  178-21*  249-24
```

CROSS REFERENCE TABLE (CREF V04.00)

```
D.ECYL  134-11#  260-33*
D.END1  134-3#
D.END2  134-5#
D.END3  134-7#
D.END4  134-9#
D.HERR  134-14#  184-41*  184-42   249-24
D.IW    133-14#  133-33   174-19
D.PAT   133-34#  174-21   280-43
D.PRM   133-13#  174-13   174-35   261-34   282-9
D.RET   133-19#  133-31
D.RO    133-17#
D.SEEK  134-16#  179-15*  249-22
D.SEQ   133-21#
D.SERN  134-18#  174-5*   174-6*   174-7*   249-9    249-10   249-11
D.SERR  134-15#  178-20*  249-24
D.SIZE  134-24#  181-28   260-3    260-29   260-33*
D.TR    133-23#
D.UNIT  133-10#  168-1    180-22   180-54   181-23   185-8    186-23   186-25   202-29   211-22   248-25   249-22   254-11*  254-35*
        261-36   280-36
D.WC    133-24#  133-31
D.WCA   133-25#
D.WO    133-18#
D.XFRR  134-13#  180-43*  180-47   249-22
D.XFRW  134-12#  180-44*  181-27   181-28   249-22
D.ZERO  133-33#  174-14
D50PRG  136-21#  150-35*  217-14
DDEF    133-31#  260-28   282-11
DFPTBL  119-10#
DIAG    125-33#  219-23
DIAGMC  115-12   115-12
DIATST  268-2#
DIV10   249-13   250-14#
DIVIDE  204-39   205-7    206-16#  246-19   246-21   246-23
DLL     138-26#  165-39*  166-9*   166-10*  181-35   262-18*
DLLADR  138-31#  166-4*   166-5*   166-13   166-14
DLLDR   138-28#  165-40*  181-37   181-38
DLLNAM  138-33#  165-43*  165-44*  181-38   240-13   240-15
DLLR    138-30#  165-42*
DLLSIZ  138-32#  166-6*   166-11   166-12*  166-13*
DLLV    138-29#  165-41*
DMFRST  134-31#  150-25   150-34   154-8
DMMAIN  134-30#  214-5    214-7    214-30
DMOVRL  134-29#
DMPROG  136-20#  145-15   145-16   150-26*  151-19*  183-3    183-7    186-30   186-31   193-7    193-8    213-5    213-8    213-10
        213-11   214-4    214-6    217-10*  217-14*
DMTRLN  134-28#
DONE    160-17   187-13#
DSPSIZ  159-9    160-19#
DT.AVL  133-12#  180-24   181-25   248-28   254-11   254-35   280-36
DT.UNT  133-11#
DU.DFL  131-22#
DU.FTL  131-25#
DU.INF  131-23#
DU.QUE  131-21#
DU.SPC  131-26#  159-5
DU.TER  131-24#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
DUP     125-34#  219-20
E$END   115-12#
E$LOAD  115-12#  115-42
EF.BBR  128-27#
EF.BBU  128-28#
EF.CON  121-10#  253-1
EF.LOG  128-29#
EF.NEW  121-10#
EF.PWR  121-10#  253-9
EF.RES  121-10#  253-5    262-22
EF.SEX  128-30#
EF.STA  121-10#  253-13   262-20
EN        1-9#     1-26     1-31     1-117   115-41   116-45   120-16   137-1    139-19   150-18   151-1    156-28   212-15   212-44
        213-12   217-8    222-21   237-11   238-10   238-19   265-12   265-16   280-17   280-38   280-52   285-25   286-12   286-29
EO        1-26#  115-25   118-9    141-34   141-39   142-18   144-52   182-60   191-15   191-20   212-28   213-1    213-21   213-25
        266-5    278-16
ERR001  144-14#  263-3
ERR002  144-18#  263-8
ERR003  144-22#  263-14
ERR004  144-26#  147-7
ERR005  144-31#  243-6
ERR006  144-44#  254-29
ERR007  144-35#  150-38
ERR008  144-48#  263-20
ERR009  144-39#  153-1
ERR013  144-58#  151-31
ERR020  144-62#  231-30   245-27
ERR021  144-66#  235-5
ERR022  144-77#  234-36
ERR023  144-83#  229-40
ERR024  144-97#  229-18
ERR025  144-101# 233-50
ERR026  144-106# 269-26
ERR027  144-110# 276-19
ERR028  144-114# 271-27
ERR029  144-118# 272-5
ERR030  144-123# 156-22
ERR031  144-127# 156-40
ERR032  144-131# 159-11
ERR033  144-136# 158-21
ERR034  144-140# 216-3
ERR035  144-144# 202-24
ERR036  144-149# 222-34
ERR037  144-153# 223-5
ERR038  144-157# 267-10
ERR23A  144-86#  144-93
ERR23B  144-87   144-91#
ERR23C  144-90   144-94#
ERRBLK  136-8#
ERRC    188-35   188-42   192-9#
ERRCHR  136-11#  209-6*   209-13   209-15   209-17   209-19
ERRD    188-44   192-21#
ERRLIM  141-10#  185-8
ERRMC   160-15   184-29#
ERRME1  141-8#   188-40   190-23
ERRMES  160-14   182-24#  184-30
```

CROSS REFERENCE TABLE (CREF V04.00)

```
ERRMSG 136-8#
ERRMSL 182-65   183-1#
ERRMSX 182-57   182-67   183-20   183-22#
ERRNBR 136-8#
ERRNL  141-4#   209-11
ERRONE 141-3#   209-8
ERRRSZ 190-18   191-12#
ERRRTB 190-21   191-3#   191-12
ERRRTN 145-1#   211-40
ERRTYP 136-8#   211-31
EVL    121-10#
F$AU   115-12#
F$AUTO 115-12#  264-10   264-12
F$BGN  115-12#  115-33   120-21   121-3    144-14   144-18   144-22   144-26   144-31   144-35   144-39   144-44   144-48   144-58
       144-62   144-66   144-77   144-83   144-97   144-101  144-106  144-110  144-114  144-118  144-123  144-127  144-131  144-136
       144-140  144-144  144-149  144-153  144-157  145-1    225-10   226-18   244-5    244-11   246-38   247-3    247-10   251-8
       252-8    262-31   264-10   265-8    265-23   266-3    266-6    266-18   267-1    267-1    267-14   268-1    268-1    268-15
       269-1    269-1    269-32   270-1    270-1    272-11   273-1    273-1    273-4    274-1    274-1    274-6    275-1    275-1
       275-10   275-14   277-3    277-24   278-3    278-14   279-3    282-49   282-51   282-53   283-3    283-14   285-12   286-4
       286-4    288-11   289-1    289-3    289-3    289-12   289-14
F$CLEA 115-12#  265-8    265-21
F$DU   115-12#
F$END  115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12   115-12
       115-12   115-12   115-12#  115-33   120-21   121-3    144-16   144-20   144-24   144-28   144-33   144-37   144-41   144-46
       144-50   144-60   144-64   144-75   144-81   144-95   144-99   144-103  144-108  144-112  144-116  144-120  144-125  144-129
       144-134  144-138  144-142  144-147  144-151  144-155  144-159  145-22   225-14   226-21   244-9    244-13   246-38   247-3
       250-12   250-43   262-31   263-23   264-12   265-21   265-23   266-3    266-6    266-6    266-6    266-18   267-1    267-1
       267-14   267-14   268-1    268-1    268-15   268-15   269-1    269-1    269-32   269-32   270-1    270-1    272-11   272-11
       273-1    273-1    273-4    273-4    274-1    274-1    274-6    274-6    275-1    275-1    275-10   275-10   275-14   275-14
       277-3    277-3    277-3    277-24   277-24   278-3    278-3    278-3    278-14   278-14   279-3    279-3    279-3    282-49
       282-51   282-51   282-53   283-3    284-9    286-15   288-11   289-1    289-3    289-12   289-14
F$HARD 115-12#  283-14   284-9
F$HW   115-12#  119-10   119-18
F$INIT 115-12#  252-8    263-23
F$JMP  115-12#  250-12   250-12   262-31   266-18   282-49
F$MOD  115-12#  115-33   120-21   121-3    246-38   247-3    265-23   266-3    282-53   283-3    288-11
F$MSG  115-12#  144-14   144-16   144-18   144-20   144-22   144-24   144-26   144-28   144-31   144-33   144-35   144-37   144-39
       144-41   144-44   144-46   144-48   144-50   144-58   144-60   144-62   144-64   144-66   144-75   144-77   144-81   144-83
       144-95   144-97   144-99   144-101  144-103  144-106  144-108  144-110  144-112  144-114  144-116  144-118  144-120  144-123
       144-125  144-127  144-129  144-131  144-134  144-136  144-138  144-140  144-142  144-144  144-147  144-149  144-151  144-155
       144-155  144-157  144-159  145-1    145-22
F$PROT 115-12#  251-8    251-14
F$PWR  115-12#
F$RPT  115-12#  247-10   250-43
F$SEG  115-12#
F$SOFT 115-12#  285-12   286-15
F$SRV  115-12#  225-10   225-14   226-18   226-21   244-5    244-9    244-11   244-13
F$SUB  115-12#  267-1    267-14   268-1    268-15   269-1    269-32   270-1    272-11   273-1    273-4    274-1    274-6    275-1
       275-10
F$SW   115-12#  120-10   120-19
F$TEST 115-12#  266-6    275-14   277-3    277-24   278-3    278-14   279-3    282-51
FDATA  137-8#   240-5    240-15   241-2    241-9    242-14*  242-17*
FFREE  136-13#  144-84   148-16   148-20*  150-16*  163-28   163-40   163-44   166-4    166-12   166-14*  214-15   229-3    229-32
       231-17   255-2*   255-3    256-3    262-14   275-2    275-9*   277-11   277-18*
FILOPN 137-9#   238-11   238-14*  239-26   239-29*
FMEM   136-15#  150-16   262-14*
```

```
FMEMS   136-16#  150-17   262-15*
FMERR   147-7#   148-18   228-40
FNAME   137-1#   239-28
FNUM    136-31#  150-22   154-9    183-10*  239-25*  242-8*   255-5*   280-19
FS        1-8#     1-19     1-26     1-28     1-63   115-42   116-45   137-4    139-16   150-18   151-1    212-44   217-8    237-11
        238-10   238-19   265-12   265-16   280-17   280-38   280-52
FSIZE   136-14#  148-17*  150-17*  163-29   163-42   166-6    166-11*  228-38   255-3*   262-15   275-3    275-8*   277-12   277-17*
FWORD   240-4    240-6    241-1    241-3    242-12#
G$CNTO  115-12#
G$DELM  115-12#
G$DISP  115-12#  286-4
G$EXCP  115-12#
G$HILI  115-12#
G$LOLI  115-12#
G$NO    115-12#  170-3    262-6
G$OFFS  115-12#  170-3    262-6    284-1    284-2    284-3    284-4    284-5    284-7    286-1    286-6    286-7    286-8    286-9
        286-10
G$OFSI  115-12#  170-3    262-6    284-1    284-2    284-3    284-4    284-5    284-7    286-1    286-6    286-7    286-8    286-9
        286-10
G$PRMA  115-12#  284-1    284-2
G$PRMD  115-12#  170-3    284-3    284-4    284-5    286-6    286-7
G$PRML  115-12#  262-6    284-7    286-1    286-8    286-9    286-10
G$RADA  115-12#  170-3
G$RADB  115-12#
G$RADD  115-12#  284-3    284-4    284-5    286-6    286-7
G$RADL  115-12#  262-6    284-7    286-1    286-8    286-9    286-10
G$RADO  115-12#  284-1    284-2
G$XFER  115-12#
G$YES   115-12#  284-1    284-2    284-3    284-4    284-5    284-7    286-1    286-6    286-7    286-8    286-9    286-10
GETCDN  203-16   203-18   203-28#
GETCNT  189-12   190-9    190-17   190-29   203-13#  204-15
GETCNX  203-15#  203-27
GETCXX  203-29   203-31#
GTDRVT  167-40   174-3    176-23   177-16   178-18   179-13   180-20   184-36   186-21   202-13#  211-20   254-32
GTDUST  151-25   217-6    218-5#
GTT4    154-6    154-8#
GTT450  151-49   152-44   154-7#
GTT452  151-51   152-22   154-5#
H.BRL   284-3    284-13#
H.BST   284-4    284-14#
H.CST   284-7    284-17#
H.LDR   284-5    284-15#
H.UBA   284-1    284-11#
H.VEC   284-2    284-12#
HC.BF1  125-24#  125-25   159-16
HC.BF2  125-25#  125-27   144-176  159-3    159-7*   159-8    159-35
HC.BSZ  125-13#  125-25   125-27   159-19   183-12   183-16   214-10   214-16   221-19   221-21   279-20
HC.CCT  125-19#  220-25*
HC.CEV  125-22#  125-23   219-18
HC.CMD  125-18#  125-19   125-20   149-21*
HC.CPK  125-23#  125-24   144-163  149-20   213-8*   213-9*   213-10*  213-11*  215-18*  215-19*  215-20*  215-21*  219-29*  220-17*
        221-18*  221-19*
HC.ESZ  125-11#  125-21   125-23   220-20
HC.INT  125-15#  125-16
HC.ISZ  125-9#   125-16   228-37
HC.MCT  125-17#  220-24*
```

CROSS REFERENCE TABLE (CREF V04.00)

```
HC.MEV  125-20#  125-21   220-18
HC.MPK  125-21#  125-22   144-165  149-18   149-20   151-26   158-9    158-14   158-19   213-17   215-25   217-7
HC.MSG  125-16#  125-17   125-18   149-19*  231-18
HC.PSZ  125-12#  125-22   125-24   219-19   219-25   220-20
HC.RSZ  125-10#  125-18   125-20
HC.SIZ  125-27#  149-14
HCOMM   149-14#  212-43   218-15
HM.CYL  260-27   261-25   283-26#  284-7
HO.BRL  257-16   258-3    283-22#  284-3    284-3    284-3
HO.BST  257-21   258-9    283-23#  284-4    284-4    284-4
HO.LDR  254-31   260-15   260-22   283-24#  284-5    284-5    284-5
HO.PRM  260-24   283-25#  284-7    284-7    284-7
HO.UBA  256-25   283-20#  284-1    284-1    284-1
HO.VEC  256-29   257-19   258-6    283-21#  284-2    284-2    284-2
HOE     121-10#
I$AU    115-12#
I$AUTO  115-12#  264-10#  264-12#
I$CLN   115-12#  265-8#   265-21#
I$DU    115-12#
I$HRD   283-14#  284-9#   286-4
I$INIT  115-12#  252-8#   262-31   263-23#
I$MOD   115-12#  115-33   115-33#  120-21   120-21#  121-3    121-3#   246-38   246-38#  247-3    247-3#   265-23   265-23#  266-3
        266-3#   282-53   282-53#  283-3    283-3#   288-11   288-11#
I$MSG   115-12#  144-14#  144-16#  144-18#  144-20#  144-22#  144-24#  144-26#  144-28#  144-31#  144-33#  144-35#  144-37#  144-39#
        144-41#  144-44#  144-46#  144-48#  144-50#  144-58#  144-60#  144-62#  144-64#  144-66#  144-75#  144-77#  144-81#  144-83#
        144-95#  144-97#  144-99#  144-101# 144-103# 144-106# 144-108# 144-110# 144-112# 144-114# 144-116# 144-118# 144-120# 144-123#
        144-125# 144-127# 144-129# 144-131# 144-134# 144-136# 144-138# 144-140# 144-142# 144-144# 144-147# 144-149# 144-151# 144-153#
        144-155# 144-157# 144-159# 145-1#   145-22#
I$PROT  115-12#  251-8#
I$PTAB  115-12#  289-3    289-3#   289-12   289-12#
I$PWR   115-12#
I$RPT   115-12#  247-10#  250-43#
I$SEG   115-12#  266-6    267-1    268-1    269-1    270-1    273-1    274-1    275-1    277-3    278-3    279-3
I$SETU  115-12#  289-1    289-1#   289-3    289-14   289-14#
I$SFT   285-12#  286-4    286-15#
I$SRV   115-12#  225-10#  225-14#  226-18#  226-21#  244-5#   244-9#   244-11#  244-13#
I$SUB   115-12#  266-6    267-1    267-1#   267-14   267-14#  267-14#  268-1    268-1#   268-15   268-15#  268-15#  269-1    269-1#
        269-32   269-32#  269-32#  270-1    270-1#   272-11   272-11#  272-11#  273-1    273-1#   273-4    273-4#   273-4#   274-1
        274-1#   274-6    274-6#   274-6#   275-1    275-1#   275-10   275-10#  275-10#  277-3    278-3    279-3
I$TST   115-12#  266-6    266-6#   266-18   267-1    268-1    269-1    270-1    273-1    274-1    275-1    275-14   275-14#  275-14#
        277-3    277-3#   277-24   277-24#  277-24#  278-3    278-3#   278-14   278-14#  278-14#  279-3    279-3#   282-49   282-51
        282-51#  282-51#
IBE     121-10#
ICONT   136-27#  253-4    279-11
IDU     121-10#  180-50   182-38   184-45
IER     121-10#
IFLAGS  136-26#  174-15   253-3*   253-4*   253-7*   255-6*   279-11   279-13*  280-11   282-24*  282-25*
INIT1   253-2    253-5#
INIT10  259-11#  259-14
INIT11  259-12   260-3#
INIT12  260-13#  260-19
INIT13  260-14   260-21#
INIT14  259-6    261-3#
INIT15  261-26   262-5    262-8    262-14#
INIT1A  253-6    253-9#
INIT2   253-10   253-13#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
INIT3   253-14   255-1#
INIT3L  260-30#  260-32
INIT4   256-15#  258-17
INIT5   256-21#  256-33
INIT7   257-25#  257-27
INIT8   259-5#   261-5
INITC1  254-5#   254-16
INITC2  254-9#   254-13
INITC3  254-10   254-14#
INITC4  254-21#  254-38
INITC5  254-24#  254-28
INITC6  254-34#
INITC7  254-22   254-36#
INITCC  254-25   254-31#
INITE1  254-29#  254-33
INITIM  262-21   262-24#
INITW1  261-29#  261-41
INITW2  261-32#  261-38
INITW3  261-35   261-37#
INITW4  261-33   261-39#
INITWA  141-13#  261-27
INITWB  141-14#  261-36
INITWC  140-4#   262-6
INITXX  254-39   262-17#
INTEST  270-2#
INTRCV  136-36#  244-12*  270-24*  270-33   271-9*   271-23
INTSRV  244-11#  270-19
INTST0  141-11#  270-17
INTST1  141-12#  272-7
IPADRS  137-19#  245-14   256-6    257-4
IREST   136-28#  253-7    280-11
ISR     121-10#
ISTRT   136-29#  255-6    280-11
ISTRTH  136-30#  174-15   253-3    282-25
IXE     121-10#
J$JMP   115-12#  250-12
KNO     255-19   255-33#
KPRI    262-23   262-31#
KTBASA  136-23#
KTBASO  136-24#
KTMEM   138-11#
KW.BRL  138-4#   255-21*
KW.CSR  138-3#   156-33   157-3    222-27   234-28   244-8*   246-13   255-17*  255-20*  255-25*  271-1    276-12
KW.EL   138-7#   156-35   156-38   157-5    157-8    222-29   222-32   227-54   227-55   227-56   234-30   234-33   244-6*   244-7*
        246-16   246-17   255-11*  255-12*  271-3    271-6    276-14   276-17
KW.HZ   138-6#   227-30   246-18   255-23*
KW.VEC  138-5#   255-22*  255-24
KW11I   244-5#   255-24
KWOUT.  244-8    255-10#  255-25
KYES    255-14   255-16   255-20#
L$ACP   115-42#
L$APT   115-42#
L$AUT   115-42#
L$AUTO  115-42   264-10#
L$CCP   115-42#
L$CLEA  115-42   265-8#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
L$CO    115-42#
L$DEPO  115-42#
L$DESC  115-42   139-17#
L$DESP  115-42#
L$DEVP  115-42#
L$DISP  115-42   118-9#
L$DLY   115-42#
L$DTP   115-42#
L$DTYP  115-42#
L$DUT   115-42#
L$DVTY  115-42   139-12#
L$EF    115-42#
L$ENVI  115-42#
L$ERRT  115-42   136-8#
L$ETP   115-42#
L$EXP1  115-42#
L$EXP4  115-42#
L$EXP5  115-42#
L$HARD  115-42   283-14   283-14#
L$HIME  115-42#  164-5
L$HPCP  115-42#
L$HPTP  115-42#
L$HW    115-42   119-10   119-10#
L$ICP   115-42#
L$INIT  115-42   252-8#
L$LADP  115-42#
L$LAST  115-42   288-9#   289-14
L$LOAD  115-42#
L$LUN   115-42#  152-28*  152-53*  155-28*  156-11*  202-29*  211-16*  266-11*
L$MREV  115-42#
L$NAME  115-42#
L$PRIO  115-42#
L$PROT  115-42   251-8#
L$PRT   115-42#
L$REPP  115-42#
L$REV   115-42#
L$RPT   115-42   247-10#
L$SOFT  115-42   285-12   285-12#
L$SPC   115-42#
L$SPCP  115-42#
L$SPTP  115-42#
L$STA   115-42#
L$SW    115-42   120-10   120-10#
L$TEST  115-42#
L$TIML  115-42#
L$UNIT  115-42#  254-37   258-16   261-4
L10000  119-10   119-18#
L10001  120-10   120-19#
L10002  144-16#
L10003  144-20#
L10004  144-24#
L10005  144-28#
L10006  144-33#
L10007  144-37#
L10010  144-41#
L10011  144-46#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
L10012 144-50#
L10013 144-60#
L10014 144-64#
L10015 144-75#
L10016 144-81#
L10017 144-95#
L10020 144-99#
L10021 144-103#
L10022 144-108#
L10023 144-112#
L10024 144-116#
L10025 144-120#
L10026 144-125#
L10027 144-129#
L10030 144-134#
L10031 144-138#
L10032 144-142#
L10033 144-147#
L10034 144-151#
L10035 144-155#
L10036 144-159#
L10037 145-22#
L10040 225-14#
L10041 226-21#
L10042 244-9#
L10043 244-13#
L10044 250-12    250-43#
L10046 262-31    263-23#
L10047 264-12#
L10050 265-21#
L10051 [illegible]6-18    275-14#
L10052 267-14#
L10053 268-15#
L10054 269-32#
L10055 272-11#
L10056 273-4#
L10057 274-6#
L10060 275-10#
L10061 277-24#
L10062 278-14#
L10063 282-49    282-51#
L10064 283-14    284-9#
L10065 285-12    286-15#
L10066 289-3#
L10070 289-3     289-12#
LBUFE  138-22#   183-9*    183-13
LBUFN  138-21#   183-6*    183-11   183-12*  183-13   183-21*  279-21
LBUFS  138-20#   150-15*   183-1    183-5*   279-14   279-17*
LDDM   155-22#   155-37
LDNEXT 155-30    155-33    155-35#
LOAD   214-23    215-15#
LOADB  213-6#    214-31
LOADDM 152-37    152-54    155-32   212-14#
LOADE1 213-18    215-26    216-3#
LOADER 212-42    214-24    215-24   216-4#   218-14
LOADT1 213-3     214-4#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
LOE     121-10#
LOG     125-32#
LOGCHK  279-15   279-26#
LOGM1   141-29#  279-16
LOGM2   141-30#  279-23
LOGM3   141-31#  279-28
LOGOUT  279-18#  279-22
LOT     121-10#
LPNT    210-6    210-8    210-10   210-12#  246-24   246-28   246-33
LPNTB   144-15   144-19   144-23   144-27   144-32   144-36   144-40   144-45   144-49   144-59   144-63   144-74   144-79   144-84
        144-88   144-94   144-98   144-102  144-107  144-111  144-115  144-119  144-124  144-128  144-132  144-145  144-150  144-154
        144-158  144-161  144-167  144-174  144-178  145-8    196-5    197-5    210-7#
LPNTF   167-35   168-1    171-32   188-40   190-23   210-5#   255-18   261-27   279-16   279-23   279-28   281-5
LPNTS   210-11#  247-13   248-8
LPNTX   180-54   180-56   185-8    186-25   210-9#   270-17   272-7
LT11    214-12   214-14#
LT1L1   214-10#  214-29
LT1L2   214-18#  214-20
M1        1-12#    1-22     1-25     1-46     1-366    115-45   139-28   265-16   284-6    284-16   286-2    286-18
M2        1-13#    1-23     1-25     1-49     1-450    115-46   133-28   134-19   139-31   142-12   142-65   144-30   144-105  163-27
        165-38   167-29   174-8    178-11   178-22   180-29   180-59   182-25   182-40   182-68   182-73   185-1    185-10   207-1
        211-25   227-6    227-17   227-31   227-63   227-72   247-17   248-12   248-26   249-25   250-5    250-36   255-27   261-7
        262-1    262-25   282-26   282-34   286-3    286-19   286-39
M3        1-16#    1-22     1-25     1-52     1-727    115-49   139-40   265-16   284-6    284-16   286-2    286-18
M4        1-17#    1-23     1-25     1-55     1-811    115-50   133-28   134-19   139-43   142-12   142-65   144-30   144-105  163-27
        165-38   167-29   174-8    178-11   178-22   180-29   180-59   182-25   182-40   182-68   182-73   185-1    185-10   207-1
        211-25   227-6    227-17   227-31   227-63   227-72   247-17   248-12   248-26   249-25   250-5    250-36   255-27   261-7
        262-1    262-25   282-26   282-34   286-3    286-19   286-39
MC        1-25#  132-31   143-9    155-23   230-4
MD        1-22#    1-24   118-10   141-34   150-51   182-60   191-15   213-1    213-25   266-5
MD.CMP  128-4#
MD.CWB  128-22#
MD.ERR  128-6#
MD.EXP  128-5#
MD.FEU  128-16#
MD.IMF  128-20#
MD.NXU  128-18#
MD.PRI  128-23#
MD.RIP  128-19#
MD.SCH  128-7#
MD.SCL  128-8#
MD.SEC  128-9#
MD.SEQ  128-14#
MD.SER  128-10#
MD.SPD  128-15#
MD.SSH  128-11#
MD.SWP  128-21#
MD.VOL  128-17#
MD.WBN  128-12#
MD.WBV  128-13#
MEMFIL  163-33#  163-36
MESSAG  160-16   186-18#
MESSG   141-16#  180-54   186-25
MLDRER  260-17   263-7#
MM        1-24#  286-34
MSCP    125-31#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
MSGPKL 144-178# 144-181
MSGPKT 144-133  144-146  144-174#
MX       1-23#    1-24   118-11   141-39   150-51   191-20   248-1    278-16   279-5    280-4    282-27
MXFERE 180-21   180-58#
MXFERP 141-9#   180-56
MXFERX 180-26   180-46   180-48   180-51   180-57#
NCON   145-18   145-20#
NCONF  188-37   188-42#
NCONS  188-36#  188-39
NEWTAB 256-22   257-3#
NOCLOC 141-28#  255-18
NXMAD  138-10#  225-12*  231-23*  231-28   245-10*  245-18   267-2*   267-8
NXMI   225-10#  231-24   245-11   267-3
NXTTAB 256-16   257-30   258-10   258-15#
O$APTS 115-12#  115-42
O$AU   115-12#  115-42
O$BGNR 115-12#  115-39#  115-42
O$BGNS 115-12#  115-39#  115-42
O$DU   115-12#  115-42
O$ERRT 115-12#  115-39#  115-42
O$GNSW 115-12#  115-39#  115-42
O$POIN 115-12#  115-39   115-39#  115-39#  115-39#  115-39#  115-39#  115-42
O$SETU 115-12#  115-39#  115-42   288-9
OP.ABO 127-3#
OP.ACC 127-4#
OP.AVA 127-22#
OP.AVL 127-5#
OP.CCD 127-6#
OP.CMP 127-7#
OP.DUP 127-23#
OP.ELP 127-29#
OP.END 127-20#  158-5    158-8
OP.ERS 127-8#
OP.ESP 127-28#  213-6
OP.FLU 127-9#
OP.GCS 127-10#
OP.GSS 127-27#  218-16
OP.GUS 127-11#
OP.MRD 127-18#
OP.MWR 127-19#  215-16   219-21
OP.ONL 127-12#
OP.RD  127-13#
OP.RLC 127-25#
OP.RPL 127-14#
OP.RSD 127-31#  158-8    159-33
OP.SCC 127-15#
OP.SEX 127-21#
OP.SHC 127-24#
OP.SSD 127-30#  158-5    159-14
OP.SUC 127-16#
OP.WR  127-17#
OSTRE  188-34   188-41   188-46#
OSTRNG 145-21   186-35   188-33#  188-45   193-9    198-6    199-6    200-6    210-17
P.BCNT 129-21#  130-9#   213-9*   215-19*  221-19*
P.BUFF 129-22#
P.CMST 130-14#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
P.CNCL 130-48#
P.CNTF 129-40#  130-46#
P.CNTI 130-49#
P.CPSP 129-34#
P.CRF  129-17#  130-4#   158-19   220-17*
P.CTMO 130-47#
P.CYL  130-26#
P.DEXT 130-52#  151-26   217-7
P.DFLG 130-53#
P.DMDT 129-50#
P.DPRG 130-54#
P.DTMO 130-55#
P.ELGF 129-32#
P.FBBK 130-10#
P.FLGS 130-7#
P.GRP  130-25#
P.HSTI 129-31#  130-19#  130-35#
P.HTMO 129-41#
P.LBN  129-24#
P.MEDI 130-21#  130-37#
P.MLUN 130-17#  130-33#
P.MOD  129-20#
P.OPCD 129-19#  130-6#   158-9    219-29*
P.OTRF 129-27#  130-13#
P.OVRL 129-51#  213-10*  213-11*
P.RBN  129-36#
P.RBNS 130-28#
P.RCTC 130-29#
P.RCTS 130-27#
P.RGID 129-46#  215-21*
P.RGOF 129-47#  215-20*
P.SHST 130-23#  130-39#
P.SHUN 129-33#  130-22#  130-38#
P.STS  130-8#   158-14   213-17   215-25
P.TIME 129-43#
P.TRCK 130-24#
P.UADR 129-23#  213-8*   215-18*  221-18*
P.UNCL 130-40#
P.UNFL 129-30#  130-18#  130-34#
P.UNIT 129-18#  130-5#
P.UNSZ 130-41#
P.UNTI 130-20#  130-36#
P.USEF 129-42#
P.VRSN 129-39#  130-45#
P.VSER 130-42#
PAT16C 137-28#  172-19   280-51   280-57
PAT16W 137-29#  280-57
PB     209-15#  210-7
PF     136-10   209-13#  210-5
PNT    121-10#
PNT.CT 144-15   144-15   144-15#  144-15#  144-19   144-19   144-19#  144-19#  144-23   144-23   144-23#  144-23#  144-27   144-27#
       144-32   144-32#  144-36   144-36#  144-40   144-40#  144-45   144-45#  144-49   144-49   144-49#  144-49#  144-59   144-59#
       144-63   144-63#  144-74   144-74   144-74   144-74   144-74#  144-74#  144-74#  144-74#  144-79   144-79   144-79   144-79#
       144-79#  144-79#  144-84   144-84   144-84   144-84#  144-84#  144-84#  144-88   144-88   144-88   144-88#  144-88#  144-88#
       144-94   144-94#  144-98   144-98   144-98#  144-98#  144-102  144-102  144-102  144-102#  144-102#  144-102#  144-107  144-107
       144-107  144-107#  144-107#  144-107#  144-111  144-111  144-111#  144-111#  144-115  144-115#  144-119  144-119  144-119#  144-119#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
RPTDT   248-23#  250-2
RPTDTN  248-27   250-1#
RPTMD2  249-24   250-35#
RPTMSD  249-22   250-34#
RPTMSG  247-13   250-31#
RPTMSH  248-8    250-32#
RPTXX   248-7    250-11#
RSPDRP  156-23   156-41   157-21#  158-22   159-12   159-24
RSPDSP  159-23   160-3#   160-19
RSPERR  158-10   158-15   158-21#
RSPIN   156-13   158-5#
RSPMWR  158-7    158-9#
RSPNRP  157-4    157-7    157-9    157-14#
RSPNTO  156-34   156-37   156-39   156-42#
RSPNXT  156-10   157-3#   157-23   159-47
RSPOU   156-15   158-27#
RSPOU2  159-31   159-40#
RSPOU3  159-38   159-41#
RSPOUT  158-27   159-29#
RSPPT2  159-3#
RSPPT3  159-10   159-14#
RSPPTW  158-20   158-26#
RSPRPT  157-6    157-10#
RSPTM   156-21   156-27#
RSPTMO  156-36   156-40#
RUNDM   155-15#  166-17   275-5    277-14   278-10
RWORDT  241-5    242-1#
RWRDE1  239-37   240-10   241-28   242-4    243-6#
S$LSYM  115-12#  119-18#  120-19#  144-16#  144-20#  144-24#  144-28#  144-33#  144-37#  144-41#  144-46#  144-50#  144-60#  144-64#
        144-75#  144-81#  144-95#  144-99#  144-103# 144-108# 144-112# 144-116# 144-120# 144-125# 144-129# 144-134# 144-138# 144-142#
        144-147# 144-151# 144-155# 144-159# 145-22#  170-3    170-3    170-3    170-3#   225-14#  226-21#  244-9#   244-13#  250-43#
        262-6    262-6    262-6    262-6#   263-23#  264-12#  265-21#  267-14#  268-15#  269-32#  272-11#  273-4#   274-6#   275-10#
        275-14#  277-24#  278-14#  282-51#  284-9#   286-15#
S.EL    286-6    286-23#
S.IW    286-9    286-26#
S.LOG   286-10   286-27#
S.MAN   286-1    286-17#
S.MES   286-4    286-20#
S.SSF   286-8    286-25#
S.XL    286-7    286-24#
SA.A2   123-18#
SA.BST  124-23#
SA.CM1  123-32#  274-2
SA.CMD  123-27#
SA.CME  123-37#
SA.CNT  124-16#  233-38   233-39
SA.CTP  123-40#
SA.DI   123-19#  233-14
SA.ERC  123-13#
SA.ERR  123-9#   144-78   234-19
SA.GO   124-21#  230-15
SA.INE  124-4#
SA.INT  123-25#  270-23
SA.LFC  124-22#
SA.MCV  124-15#  233-38
SA.MS1  123-31#  274-2
```

CROSS REFERENCE TABLE (CREF V04.00)

```
SA.MSE  123-36#
SA.MSG  123-26#
SA.NV   123-17#
SA.NVE  124-5#
SA.PRG  123-44#
SA.S1   123-5#   232-3    233-14   268-8    269-9    270-26
SA.S2   123-6#   233-23   268-12
SA.S3   123-7#   233-31
SA.S4   123-8#   232-13   233-38
SA.STE  123-39#
SA.STP  123-29#  231-15   268-11   269-13   269-16   270-23   274-2
SA.TST  124-11#  233-8
SA.VCE  124-3#
SA.VEC  123-24#
SA.WRP  123-28#  269-13   269-16
SAMTAB  256-26   258-3#
SAMVEC  256-31   263-19#
SEKERE  179-14   179-17#
SETDPR  212-47   217-5#
SETO0   227-45#  227-50
SETO1   227-46   227-48#
SETO2   227-54#  227-57
SETTO   159-46   222-15   227-27#  234-23   247-21   255-31   262-29   270-32   276-8
SFPTBL  120-10#  167-30   174-17   180-45   180-47   182-56   182-64   184-42   279-26   280-13   282-43
SM.IW   174-17   285-24#  286-9
SM.LOG  182-64   279-26   282-43   285-23#  286-10
SM.MAN  167-30   280-13   285-21#  286-1
SM.SSF  182-56   285-22#  286-8
SNDC1   220-21#  220-23
SNDCMD  159-42   213-15   215-22   218-18   220-14#
SO.BIT  167-30   174-17   182-56   182-64   279-26   280-13   282-43   285-20#  286-1    286-1    286-1    286-8    286-8    286-8
        286-9    286-9    286-9    286-10   286-10   286-10
SO.EL   184-42   285-18#  286-6    286-6    286-6
SO.XL   180-45   180-47   285-19#  286-7    286-7    286-7
ST.ABO  131-7#
ST.AVL  131-9#
ST.CMD  131-6#
ST.CMP  131-12#
ST.CNT  131-15#
ST.DAT  131-13#
ST.DIA  131-17#
ST.DRV  131-16#
ST.HST  131-14#
ST.MFE  131-10#
ST.MSK  131-3#   158-14   213-17   215-25
ST.OFL  131-8#
ST.SUB  131-4#
ST.SUC  131-5#
ST.WPR  131-11#
STIME   138-8#   157-5    157-8    247-16   255-26   262-24
STLDDM  151-52   155-20#
STORAG  117-8#   150-25   150-26   150-29   150-30   151-19   154-8    217-10   241-11   241-14   280-39   280-53
STOSIZ  117-1#   117-8    241-14
SVCGBL  115-12#  115-17#  115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   118-9    119-10
```

```
       119-10  120-10  120-10  136-8   139-12  139-17  144-11# 144-14  144-14  144-14  144-18  144-18  144-18  144-22
       144-22  144-22  144-26  144-26  144-26  144-31  144-31  144-31  144-35  144-35  144-35  144-39  144-39  144-39
       144-44  144-44  144-44  144-48  144-48  144-48  144-58  144-58  144-58  144-62  144-62  144-62  144-66  144-66
       144-66  144-77  144-77  144-77  144-83  144-83  144-83  144-97  144-97  144-97  144-101 144-101 144-101 144-106
       144-106 144-106 144-110 144-110 144-110 144-114 144-114 144-114 144-118 144-118 144-118 144-123 144-123 144-123
       144-127 144-127 144-127 144-131 144-131 144-131 144-136 144-136 144-136 144-140 144-140 144-140 144-144 144-144
       144-144 144-149 144-149 144-149 144-153 144-153 144-153 144-157 144-157 144-157 145-1   145-1   145-1   146-4#
       225-10  226-18  244-5   244-11  247-10  251-8   252-8   264-10  265-8   283-14  285-12  288-9   288-9   288-9
       288-9#
SVCINS 115-12# 115-14# 115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42  115-42
       115-42  115-42  115-42  118-9   118-9   118-9   118-9   118-9   118-9   118-9   118-9   118-9   118-9   118-9
       118-9   118-9   118-9   118-9   119-10  119-10  119-10  120-10  120-10  120-10  139-12  139-12  139-12  139-12
       139-12  139-12  139-17  139-17  139-17  139-17  139-17  139-17  144-8#  144-16  144-20  144-24  144-28  144-33
       144-37  144-41  144-46  144-50  144-60  144-64  144-75  144-80  144-80  144-80  144-80  144-80  144-81  144-95
       144-99  144-103 144-108 144-112 144-116 144-120 144-125 144-129 144-134 144-138 144-142 144-147 144-151 144-155
       144-159 145-22  146-1#  147-7   147-7   147-7   147-7   147-7   147-7   147-7   147-7   147-7   147-7   147-7
       147-7   147-8   147-8   147-8   150-38  150-38  150-38  150-38  150-38  150-38  150-38  150-38  150-38  150-38
       150-38  150-38  150-39  150-39  150-39  151-31  151-31  151-31  151-31  151-31  151-31  151-31  151-31  151-31
       151-31  151-31  151-31  151-32  151-32  151-32  153-1   153-1   153-1   153-1   153-1   153-1   153-1   153-1
       153-1   153-1   153-1   153-1   156-22  156-22  156-22  156-22  156-22  156-22  156-22  156-22  156-22  156-22
       156-22  156-22  156-40  156-40  156-40  156-40  156-40  156-40  156-40  156-40  156-40  156-40  156-40  156-40
       156-43  156-43  156-43  157-10  157-10  157-10  158-21  158-21  158-21  158-21  158-21  158-21  158-21  158-21
       158-21  158-21  158-21  158-21  159-11  159-11  159-11  159-11  159-11  159-11  159-11  159-11  159-11  159-11
       159-11  159-11  167-33  167-33  167-33  167-34  167-34  167-34  170-3   170-3   170-3   170-3   170-3   170-3
       170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3   170-3
       170-3   170-3   170-3   170-3   180-49  180-49  180-49  182-37  182-37  182-37  184-44  184-44  184-44  202-24
       202-24  202-24  202-24  202-24  202-24  202-24  202-24  202-24  202-24  202-24  202-24  209-13  209-13  209-13
       209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13  209-13
       209-13  209-15  209-15  209-15  209-15  209-15  209-15  209-15  209-15  209-15  209-15  209-15  209-15  209-15
       209-15  209-15  209-15  209-15  209-15  209-17  209-17  209-17  209-17  209-17  209-17  209-17  209-17  209-17
       209-17  209-17  209-17  209-17  209-17  209-17  209-17  209-17  209-17  209-19  209-19  209-19  209-19  209-19
       209-19  209-19  209-19  209-19  209-19  209-19  209-19  209-19  209-19  209-19  209-19  209-19  209-19  211-41
       211-41  211-41  212-37  212-37  212-37  212-37  212-37  212-37  212-37  212-37  212-37  212-37  212-37  212-37
       212-37  212-37  212-37  212-37  212-37  212-37  216-3   216-3   216-3   216-3   216-3   216-3   216-3   216-3
       216-3   216-3   216-3   216-3   218-9   218-9   218-9   218-9   218-9   218-9   218-9   218-9   218-9   218-9
       218-9   218-9   218-9   218-9   218-9   218-9   218-9   218-9   222-26  222-26  222-26  222-34  222-34  222-34
       222-34  222-34  222-34  222-34  222-34  222-34  222-34  222-34  222-34  223-5   223-5   223-5   223-5   223-5
       223-5   223-5   223-5   223-5   223-5   223-5   223-5   225-14  225-14  225-14  226-21  226-21  226-21  229-18
       229-18  229-18  229-18  229-18  229-18  229-18  229-18  229-18  229-18  229-18  229-18  229-40  229-40  229-40
       229-40  229-40  229-40  229-40  229-40  229-40  229-40  229-40  229-40  231-13  231-13  231-13  231-24  231-24
       231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24  231-24
       231-24  231-24  231-27  231-27  231-27  231-27  231-27  231-27  231-30  231-30  231-30  231-30  231-30  231-30
       231-30  231-30  231-30  231-30  231-30  231-30  233-50  233-50  233-50  233-50  233-50  233-50  233-50  233-50
       233-50  233-50  233-50  233-50  234-27  234-27  234-27  234-36  234-36  234-36  234-36  234-36  234-36  234-36
       234-36  234-36  234-36  234-36  234-36  235-5   235-5   235-5   235-5   235-5   235-5   235-5   235-5   235-5
       235-5   235-5   235-5   238-13  238-13  238-13  239-28  239-28  239-28  239-28  239-28  239-28  239-31  239-31
       239-31  239-33  239-33  239-33  239-33  239-33  239-33  239-38  239-38  239-38  240-7   240-7   240-7   241-17
```

CROSS REFERENCE TABLE (CREF V04.00)

241-17	241-17	241-24	241-24	241-24	242-1	242-1	242-1	242-12	242-12	242-12	242-15	242-15	242-15	
243-6	243-6	243-6	243-6	243-6	243-6	243-6	243-6	243-6	243-6	243-6	243-6	243-7	243-7	
243-7	244-9	244-9	244-9	244-13	244-13	244-13	245-11	245-11	245-11	245-11	245-11	245-11	245-11	
245-11	245-11	245-11	245-11	245-11	245-11	245-11	245-11	245-11	245-11	245-11	245-12	245-12	245-12	
245-27	245-27	245-27	245-27	245-27	245-27	245-27	245-27	245-27	245-27	245-27	245-27	245-29	245-29	
245-29	245-30	245-30	245-30	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	
249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	
249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-22	249-24	249-24	249-24	249-24	249-24	
249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	249-24	
249-24	249-24	249-24	249-24	249-24	250-12	250-12	250-12	250-12	250-12	250-12	250-43	250-43	250-43	
253-1	253-1	253-1	253-1	253-1	253-1	253-2	253-2	253-2	253-5	253-5	253-5	253-5	253-5	
253-5	253-6	253-6	253-6	253-9	253-9	253-9	253-9	253-9	253-9	253-10	253-10	253-10	253-12	
253-12	253-12	253-13	253-13	253-13	253-13	253-13	253-13	253-14	253-14	253-14	254-21	254-21	254-21	
254-21	254-21	254-21	254-22	254-22	254-22	254-29	254-29	254-29	254-29	254-29	254-29	254-29	254-29	
254-29	254-29	254-29	254-29	254-30	254-30	254-30	255-2	255-2	255-2	255-2	255-2	255-2	255-13	
255-13	255-13	255-13	255-13	255-13	255-14	255-14	255-14	255-15	255-15	255-15	255-15	255-15	255-15	
255-16	255-16	255-16	255-24	255-24	255-24	255-24	255-24	255-24	255-24	255-24	255-24	255-24	255-24	
255-24	255-24	255-24	255-24	255-24	255-24	255-24	256-15	256-15	256-15	256-15	256-15	256-15	256-16	
256-16	256-16	259-5	259-5	259-5	259-5	259-5	259-5	259-6	259-6	259-6	261-36	261-36	261-36	
261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	261-36	
261-36	261-36	261-36	261-36	261-36	261-36	261-36	262-4	262-4	262-4	262-5	262-5	262-5	262-6	
262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	262-6	
262-6	262-6	262-6	262-9	262-9	262-9	262-17	262-17	262-17	262-17	262-17	262-17	262-20	262-20	
262-20	262-20	262-20	262-20	262-21	262-21	262-21	262-22	262-22	262-22	262-22	262-22	262-22	262-23	
262-23	262-23	262-31	262-31	262-31	262-31	262-31	262-31	263-3	263-3	263-3	263-3	263-3	263-3	
263-3	263-3	263-3	263-3	263-3	263-3	263-4	263-4	263-4	263-8	263-8	263-8	263-8	263-8	
263-8	263-8	263-8	263-8	263-8	263-8	263-8	263-9	263-9	263-9	263-14	263-14	263-14	263-14	
263-14	263-14	263-14	263-14	263-14	263-14	263-14	263-14	263-15	263-15	263-15	263-20	263-20	263-20	
263-20	263-20	263-20	263-20	263-20	263-20	263-20	263-20	263-20	263-21	263-21	263-21	263-23	263-23	
263-23	264-12	264-12	264-12	265-21	265-21	265-21	266-18	266-18	266-18	266-18	266-18	266-18	267-1	
267-1	267-1	267-3	267-3	267-3	267-3	267-3	267-3	267-3	267-3	267-3	267-3	267-3	267-3	
267-3	267-3	267-3	267-3	267-3	267-3	267-7	267-7	267-7	267-7	267-7	267-7	267-10	267-10	
267-10	267-10	267-10	267-10	267-10	267-10	267-10	267-10	267-10	267-10	267-11	267-11	267-11	267-14	
267-14	267-14	268-1	268-1	268-1	268-15	268-15	268-15	269-1	269-1	269-1	269-26	269-26	269-26	
269-26	269-26	269-26	269-26	269-26	269-26	269-26	269-26	269-26	269-32	269-32	269-32	270-1	270-1	
270-1	270-19	270-19	270-19	270-19	270-19	270-19	270-19	270-19	270-19	270-19	270-19	270-19	270-19	
270-19	270-19	270-19	270-19	270-19	270-20	270-20	270-20	270-20	270-20	270-20	270-35	270-35	270-35	
271-10	271-10	271-10	271-10	271-10	271-10	271-21	271-21	271-21	271-21	271-21	271-21	271-27	271-27	
271-27	271-27	271-27	271-27	271-27	271-27	271-27	271-27	271-27	271-27	272-1	272-1	272-1	272-1	
272-1	272-1	272-5	272-5	272-5	272-5	272-5	272-5	272-5	272-5	272-5	272-5	272-5	272-5	
272-10	272-10	272-10	272-10	272-10	272-10	272-11	272-11	272-11	273-1	273-1	273-1	273-4	273-4	
273-4	274-1	274-1	274-1	274-6	274-6	274-6	275-1	275-1	275-1	275-10	275-10	275-10	275-14	
275-14	275-14	276-11	276-11	276-11	276-19	276-19	276-19	276-19	276-19	276-19	276-19	276-19	276-19	
276-19	276-19	276-19	277-24	277-24	277-24	278-14	278-14	278-14	280-15	280-15	280-15	280-16	280-16	
280-16	282-45	282-45	282-45	282-48	282-48	282-48	282-49	282-49	282-49	282-49	282-49	282-49	282-51	
282-51	282-51	283-14	283-14	283-14	284-1	284-1	284-1	284-1	284-1	284-1	284-1	284-1	284-1	
284-1	284-1	284-1	284-2	284-2	284-2	284-2	284-2	284-2	284-2	284-2	284-2	284-2	284-2	
284-2	284-3	284-3	284-3	284-3	284-3	284-3	284-3	284-3	284-3	284-3	284-3	284-3	284-3	
284-3	284-3	284-4	284-4	284-4	284-4	284-4	284-4	284-4	284-4	284-4	284-4	284-4	284-4	
284-4	284-4	284-4	284-5	284-5	284-5	284-5	284-5	284-5	284-5	284-5	284-5	284-5	284-5	
284-5	284-5	284-5	284-5	284-7	284-7	284-7	284-7	284-7	284-7	284-7	284-7	284-7	284-9	
284-9	284-9	285-12	285-12	285-12	286-1	286-1	286-1	286-1	286-1	286-1	286-1	286-1	286-1	
286-4	286-4	286-4	286-4	286-4	286-4	286-6	286-6	286-6	286-6	286-6	286-6	286-6	286-6	
286-6	286-6	286-6	286-6	286-6	286-6	286-6	286-7	286-7	286-7	286-7	286-7	286-7	286-7	
286-7	286-7	286-7	286-7	286-7	286-7	286-7	286-7	286-8	286-8	286-8	286-8	286-8	286-8	
286-8	286-8	286-8	286-9	286-9	286-9	286-9	286-9	286-9	286-9	286-9	286-9	286-10	286-10	

CROSS REFERENCE TABLE (CREF V04.00)

	286-10	286-10	286-10	286-10	286-10	286-10	286-10	286-15	286-15	286-15	288-9	288-9	288-9	288-9
	288-9	288-9	288-9	288-9	288-9	289-3	289-3	289-3	289-3	289-3	289-3			
SVCSUB	115-12#	115-16#	144-10#	146-3#	267-1	268-1	269-1	270-1	273-1	274-1	275-1			
SVCTAG	115-12#	115-18#	119-18	120-19	144-12#	144-16	144-16	144-16	144-20	144-20	144-20	144-24	144-24	144-24
	144-28	144-28	144-28	144-33	144-33	144-33	144-37	144-37	144-37	144-41	144-41	144-41	144-46	144-46
	144-46	144-50	144-50	144-50	144-60	144-60	144-60	144-64	144-64	144-64	144-75	144-75	144-75	144-81
	144-81	144-81	144-95	144-95	144-95	144-99	144-99	144-99	144-103	144-103	144-103	144-108	144-108	144-108
	144-112	144-112	144-112	144-116	144-116	144-116	144-120	144-120	144-120	144-125	144-125	144-125	144-129	144-129
	144-129	144-134	144-134	144-134	144-138	144-138	144-138	144-142	144-142	144-142	144-147	144-147	144-147	144-151
	144-151	144-151	144-155	144-155	144-155	144-159	144-159	144-159	145-22	145-22	145-22	146-5#	170-3	225-14
	226-21	244-9	244-13	250-43	262-6	263-23	264-12	265-21	267-14	268-15	269-32	272-11	273-4	274-6
	275-10	275-14	277-24	278-14	282-51	284-9	286-15	289-3	289-12					
SVCTST	115-12#	115-15#	144-9#	146-2#	266-6	277-3	278-3	279-3						
T$$AUT	264-10#	264-12												
T$$CLE	265-8#	265-21												
T$$DAT	289-3	289-3#	289-12											
T$$HAR	283-14	283-14#	284-9											
T$$HW	119-10	119-10#	119-18											
T$$INI	252-8#	262-31	263-23											
T$$MSG	144-14#	144-16	144-18#	144-20	144-22#	144-24	144-26#	144-28	144-31#	144-53	144-35#	144-37	144-39#	144-41
	144-44#	144-46	144-48#	144-50	144-58#	144-60	144-62#	144-64	144-66#	144-75	144-77#	144-81	144-83#	144-95
	144-97#	144-99	144-101#	144-103	144-106#	144-108	144-110#	144-112	144-114#	144-116	144-118#	144-120	144-123#	144-125
	144-127#	144-129	144-131#	144-134	144-136#	144-138	144-140#	144-142	144-144#	144-147	144-149#	144-151	144-153#	144-155
	144-157#	144-159	145-1#	145-22										
T$$PC	289-1#	289-14												
T$$PRO	251-8#													
T$$PTA	289-1#	289-3	289-3#											
T$$RPT	247-10#	250-12	250-43											
T$$SOF	285-12	285-12#	286-15											
T$$SRV	225-10#	225-14	226-18#	226-21	244-5#	244-9	244-11#	244-13						
T$$SUB	267-1#	267-14	268-1#	268-15	269-1#	269-32	270-1#	272-11	273-1#	273-4	274-1#	274-6	275-1#	275-10
T$$SW	120-10	120-10#	120-19											
T$$TES	266-6#	266-18	275-14	277-3#	277-24	278-3#	278-14	279-3#	282-49	282-51				
T$ARGC	115-42	115-42	115-42	115-42	115-42	115-42	115-42	115-42	115-42	115-42	115-42	115-42#	115-42#	115-42#
	115-42#	115-42#	115-42#	144-80	144-80	144-80#	209-13	209-13	209-13	209-13#	209-13#	209-15	209-15	209-15
	209-15#	209-15#	209-17	209-17	209-17	209-17#	209-17#	209-19	209-19	209-19	209-19#	209-19#	249-22	249-22
	249-22	249-22	249-22	249-22	249-22	249-22	249-22#	249-22#	249-22#	249-22#	249-22#	249-22#	249-22#	249-24
	249-24	249-24	249-24	249-24	249-24#	249-24#	249-24#	249-24#	261-36	261-36	261-36	261-36	261-36	261-36#
	261-36#	261-36#	261-36#											
T$CODE	170-3	170-3	170-3	170-3#	170-3#	170-3#	262-6	262-6	262-6	262-6#	262-6#	262-6#	284-1	284-1
	284-1	284-1#	284-1#	284-1#	284-2	284-2	284-2	284-2#	284-2#	284-2#	284-3	284-3	284-3	284-3#
	284-3#	284-3#	284-4	284-4	284-4	284-4#	284-4#	284-4#	284-5	284-5	284-5	284-5#	284-5#	284-5#
	284-7	284-7	284-7	284-7#	284-7#	284-7#	286-1	286-1	286-1	286-1#	286-1#	286-1#	286-4	286-4#
	286-6	286-6	286-6	286-6#	286-6#	286-6#	286-7	286-7	286-7	286-7#	286-7#	286-7#	286-8	286-8
	286-8	286-8#	286-8#	286-8#	286-9	286-9	286-9	286-9#	286-9#	286-9#	286-10	286-10	286-10	286-10#
	286-10#	286-10#												
T$ERRN	115-12#	147-7	147-7#	150-38	150-38#	151-31	151-31#	153-1	153-1#	156-22	156-22#	156-40	156-40#	158-21
	158-21#	159-11	159-11#	202-24	202-24#	216-3	216-3#	222-34	222-34#	223-5	223-5#	229-18	229-18#	229-40
	229-40#	231-30	231-30#	233-50	233-50#	234-36	234-36#	235-5	235-5#	243-6	243-6#	245-27	245-27#	254-29
	254-29#	263-3	263-3#	263-8	263-8#	263-14	263-14#	263-20	263-20#	267-10	267-10#	269-26	269-26#	271-27
	271-27#	272-5	272-5#	276-19	276-19#									
T$EXCP	170-3	170-3#	284-1	284-1#	284-2	284-2#	284-3	284-3#	284-4	284-4#	284-5	284-5#	286-6	286-6#
	286-7	286-7#												
T$FLAG	250-12	250-12#	250-12#	262-31	262-31	262-31#	262-31#	266-18	266-18	266-18#	266-18#	282-49	282-49	282-49#
	282-49#													
T$FREE	288-9	289-14#												

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

```
T$SIZE  288-9    289-14#
T$SUBN  115-12#  266-6#   267-1    267-1    267-1#   268-1    268-1    268-1#   269-1    269-1    269-1#   270-1    270-1    270-1#
        273-1    273-1    273-1#   274-1    274-1    274-1#   275-1    275-1    275-1#   277-3#   278-3#   279-3#
T$TAGL  115-12#
T$TAGN  115-12#  119-10   119-10   119-10#  120-10   120-10   120-10#  144-14   144-14   144-14#  144-18   144-18   144-18#  144-22
        144-22   144-22#  144-26   144-26   144-26#  144-31   144-31   144-31#  144-35   144-35   144-35#  144-39   144-39   144-39#
        144-44   144-44   144-44#  144-48   144-48   144-48#  144-58   144-58   144-58#  144-62   144-62   144-62#  144-66   144-66
        144-66#  144-77   144-77   144-77#  144-83   144-83   144-83#  144-97   144-97   144-97#  144-101  144-101  144-101#  144-106
        144-106  144-106#  144-110  144-110  144-110#  144-114  144-114  144-114#  144-118  144-118  144-118#  144-123  144-123  144-123#
        144-127  144-127   144-127#  144-131  144-131   144-131#  144-136  144-136  144-136#  144-140  144-140  144-140#  144-144  144-144
        144-144#  144-149  144-149   144-149#  144-153  144-153   144-153#  144-157  144-157   144-157#  145-1    145-1    145-1#   225-10
        225-10   225-10#  226-18   226-18   226-18#  244-5    244-5    244-5#   244-11   244-11   244-11#  247-10   247-10   247-10#
        251-8    251-8    251-8#   252-8    252-8    252-8#   264-10   264-10   264-10#  265-8    265-8    265-8#   266-6    266-6
        266-6#   267-1    267-1    267-1#   268-1    268-1    268-1#   269-1    269-1    269-1#   270-1    270-1    270-1#   273-1
        273-1    273-1#   274-1    274-1    274-1#   275-1    275-1    275-1#   277-3    277-3    277-3#   278-3    278-3    278-3#
        279-3    279-3    279-3#   283-14   283-14   283-14#  285-12   285-12   285-12#  289-1    289-1    289-1#   289-3    289-3
        289-3    289-3    289-3#   289-3#
T$TEMP  118-9    118-9    118-9    118-9    118-9    118-9    118-9    118-9    118-9#   118-9#   118-9#   118-9#   118-9#   119-18
        119-18#  120-19   120-19#  120-21   120-21#  144-16   144-16#  144-20   144-20#  144-24   144-24#  144-28   144-28#  144-33
        144-33#  144-37   144-37#  144-41   144-41#  144-46   144-46#  144-50   144-50#  144-60   144-60#  144-64   144-64#  144-75
        144-75#  144-81   144-81#  144-95   144-95#  144-99   144-99#  144-103  144-103#  144-108  144-108#  144-112  144-112#  144-116
        144-116#  144-120  144-120#  144-125  144-125#  144-129  144-129#  144-134  144-134#  144-138  144-138#  144-142  144-142#  144-147
        144-147#  144-151  144-151#  144-155  144-155#  144-159  144-159#  145-22   145-22#  170-3    170-3    170-3    170-3#   170-3#
        170-3#   225-14   225-14#  226-21   226-21#  244-9    244-9#   244-13   244-13#  246-38   246-38#  250-12   250-12#  250-43
        250-43#  251-14   251-14#  262-6    262-6    262-6    262-6#   262-6#   262-6#   262-31   262-31#  263-23   263-23#  264-12
        264-12#  265-21   265-21#  265-23   265-23#  266-18   266-18#  267-14   267-14#  268-15   268-15#  269-32   269-32#  272-11
        272-11#  273-4    273-4#   274-6    274-6#   275-10   275-10#  275-14   275-14#  277-24   277-24#  278-14   278-14#  282-49
        282-49#  282-51   282-51#  282-53   282-53#  284-1    284-1    284-1    284-1#   284-1#   284-1#   284-2    284-2    284-2
        284-2#   284-2#   284-2#   284-3    284-3    284-3    284-3#   284-3#   284-3#   284-4    284-4    284-4    284-4#   284-4#
        284-4#   284-5    284-5    284-5    284-5#   284-5#   284-5#   284-7    284-7    284-7    284-7#   284-7#   284-7#   284-9
        284-9#   286-1    286-1    286-1    286-1#   286-1#   286-1#   286-6    286-6    286-6    286-6#   286-6#   286-6#   286-7
        286-7    286-7    286-7#   286-7#   286-7#   286-8    286-8    286-8    286-8#   286-8#   286-8#   286-9    286-9    286-9
        286-9#   286-9#   286-9#   286-10   286-10   286-10   286-10#  286-10#  286-10#  286-15   286-15#  288-11   288-11#
T$TEST  115-12#  266-6    266-6    266-6#   267-1    268-1    269-1    270-1    273-1    274-1    275-1    277-3    277-3    277-3#
        278-3    278-3    278-3#   279-3    279-3    279-3#   288-9
T$TSTM  115-12#  144-16   144-20   144-24   144-28   144-33   144-37   144-41   144-46   144-50   144-60   144-64   144-75   144-80
        144-81   144-95   144-99   144-103  144-108  144-112  144-116  144-120  144-125  144-129  144-134  144-138  144-142  144-147
        144-151  144-155  144-159  145-22   147-7    147-8    150-38   150-39   151-31   151-32   153-1    156-22   156-40   156-43
        157-10   158-21   159-11   167-33   170-3    180-49   182-37   184-44   202-24   209-13   209-15   209-17   209-19   211-41
        212-37   216-3    218-9    222-26   222-34   223-5    229-18   229-40   231-13   231-24   231-27   231-30   233-50   234-27
        234-36   235-5    238-13   239-28   239-31   239-33   239-38   240-7    241-17   241-24   242-1    242-12   242-15   243-6
        243-7    245-11   245-12   245-27   245-29   245-30   249-22   249-24   250-43   253-1    253-5    253-9    253-12   253-13
        254-21   254-29   254-30   255-2    255-13   255-15   255-24   256-15   259-5    261-36   262-4    262-6    262-9    262-17
        262-20   262-22   262-31   263-3    263-4    263-8    263-9    263-14   263-15   263-20   263-21   263-23   264-12   265-21
        266-18   267-1    267-3    267-7    267-10   267-11   267-14   268-1    268-15   269-1    269-26   269-32   270-1    270-19
        270-20   270-35   271-10   271-21   271-27   272-1    272-5    272-10   272-11   273-1    273-4    274-1    274-6    275-1
        275-10   275-14   276-11   276-19   277-24   278-14   280-15   282-45   282-48   282-49   282-51
T$TSTS  115-12#  266-6#   277-3#   278-3#   279-3#
T1      118-9    266-6#
T1.1    267-1#
T1.2    268-1#
T1.3    269-1#
T1.4    270-1#
T1.5    273-1#
T1.6    274-1#
T1.7    275-1#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
T1GOOD 267-9     267-13#
T1MSIZ 160-3     163-26#
T1NEXT 266-10#   266-17
T1NOW  266-13    266-20#
T1SKIP 266-15#   267-12    275-12
T2     118-9     277-3#
T2CMD  160-5     167-28#
T2CMD0 167-34    167-37#
T2CMD2 167-39    169-1#
T2CMD3 170-9     171-3#
T2CMD9 167-35#
T2CMDE 170-16    171-7     171-10    171-12    171-27    171-32#   208-47
T2CMDM 167-31    167-33#
T2CMDN 171-17    171-19#
T2CMDQ 169-2     170-3#    171-33
T2CMDR 171-5     171-14#
T2CMDV 170-7     170-15#
T2CMDW 171-20    171-24    171-26#
T2CMDX 167-32    167-36    170-17    171-30#
T2CMS1 141-18#   168-1
T2CMS5 141-27#   171-32
T2DLL  160-4     165-37#
T2DR   138-15#   168-4*    170-13*   170-14
T2GND1 208-16    208-19#
T2GND2 208-20#   208-41
T2GND3 208-25    208-31#
T2GNE  208-23    208-27    208-29    208-45#
T2GNUM 170-11    170-15    171-9     171-19    171-23    171-26    208-12#   208-18
T2GNX  208-14    208-39    208-42#
T2NEXT 277-10#   277-22
T2PNT  207-7     207-9     207-16#
T2PNTB 169-10    207-12#
T2PNTD 207-26    207-28#
T2PNTO 207-21    207-23#
T2PNTW 169-5     169-7     207-4#
T2WARN 141-17#   167-35
T2WRO  138-14#   168-3*    169-6     171-15    171-29*
T2WRR  138-13#   168-2*    169-4     171-14    171-28*
T3     118-9     278-3#
T4     118-9     279-3#
T4BB1  160-8     176-22#
T4BB1E 176-24    176-30#
T4BB2  160-9     177-15#
T4BB2E 177-17    177-23#
T4CON  279-24    279-27    279-29#
T4DEF  280-14    282-3#
T4DEFA 282-4#    282-20
T4DEFB 282-7#    282-17
T4DEFC 282-13#   282-15
T4DEFD 282-16#
T4DEFE 282-8     282-18#
T4DEFW 280-16    281-5#
T4EXIT 282-44    282-48#
T4MPRM 160-6     172-18#
T4MXFR 160-12    180-18#
T4OPT7 140-3#    170-3
```

CROSS REFERENCE TABLE (CREF V04.00)

```
T4PRM1  280-31#  280-50
T4PRM2  280-34#  280-47
T4PRM3  280-37   280-44   280-46#
T4PRM4  280-35   280-48#
T4RUN   280-12   280-58   282-24#
T4SEEK  160-11   179-1#
T4SOFT  160-10   178-16#
T4STRT  279-10   279-12   279-30   280-3#
T4UPRM  160-7    174-1#
T4UPRX  174-4    174-33   174-54   175-11   175-22#
T4WAIT  279-32   282-36   282-43#  282-46
T4WARN  141-15#  281-5
TEMP    137-10#  170-3    170-4    249-2    249-22   260-9*   262-6    262-7    263-7    263-13
TIEXIT  150-20   150-23   150-36#
TINDEX  132-10#  132-12   132-12   132-12#  132-13   132-13   132-13#  132-16   132-16   132-16#  132-19   132-19   132-19#  132-20
        132-20   132-20#  132-21   132-21   132-21#  132-22   132-22   132-22#  132-34   132-34   132-34#  132-35   132-35   132-35#
        132-36   132-36   132-36#  132-37   132-37   132-37#  132-38   132-38   132-38#  132-39   132-39   132-39#  132-40   132-40
        132-40#  132-41   132-41   132-41#  132-42   132-42   132-42#  132-43   132-43   132-43#  132-44   132-44   132-44#  132-45
        132-45   132-45#  132-47   133-7#   133-9    133-9    133-9#   133-10   133-10   133-10#  133-13   133-13   133-13#  133-34
        133-34   133-34#  133-35   133-35   133-35#  133-36   133-36   133-36#  133-37   133-37   133-37#  133-38   133-38   133-38#
        133-39   133-39   133-39#  133-40   133-40   133-40#  133-41   133-41   133-41#  133-42   133-42   133-42#  133-43   133-43
        133-43#  133-44   133-44   133-44#  133-45   133-45   133-45#  133-46   133-46   133-46#  133-47   133-47   133-47#  133-48
        133-48   133-48#  133-49   133-49   133-49#  133-50   133-50   133-50#  133-51   133-51   133-51#  134-1    134-1    134-1#
        134-2    134-2    134-2#   134-3    134-3    134-3#   134-4    134-4    134-4#   134-5    134-5    134-5#   134-6    134-6
        134-6#   134-7    134-7    134-7#   134-8    134-8    134-8#   134-9    134-9    134-9#   134-10   134-10   134-10#  134-11
        134-11   134-11#  134-12   134-12   134-12#  134-13   134-13   134-13#  134-14   134-14   134-14#  134-15   134-15   134-15#
        134-16   134-16   134-16#  134-17   134-17   134-17#  134-18   134-18   134-18#  134-24   283-18#  283-20   283-20   283-20#
        283-21   283-21   283-21#  283-22   283-22   283-22#  283-23   283-23   283-23#  283-24   283-24   283-24#  283-25   283-25
        283-25#  285-16#  285-18   285-18   285-18#  285-19   285-19   285-19#  285-20   285-20   285-20#
TINIT   150-13#  266-8    277-6    278-6    280-9
TINITE  150-28   150-33   150-38#  154-13   280-24
TNAMES  145-8    191-14#
TNUM    136-32#  145-2    150-13*  182-61   212-45   213-2    247-13   248-4    255-4*   265-13   279-8
TOOMER  260-20   263-13#
TSTCNT  151-23#  151-40
TSTTAB  136-19#  151-21   152-10   152-25   152-47   155-21   156-6    266-9*   266-10   266-15*  266-16   277-8*   277-20*  277-21
        278-8*   282-32*  282-37*
TY.U50  137-16#  151-35   151-43
TY.U52  137-17#  151-28   151-43   151-46
TYPCNT  137-15#  151-17*  151-28*  151-35*  151-43   151-46
U52EXT  137-12#  151-26   217-7
UAM     121-10#
UCNT    136-35#  151-20*  151-39*  152-9*   152-19*  152-24*  152-41*  152-46*  152-58*  155-20*  155-36*  156-7*   157-15*  280-29*
        280-45*
UDAI1   228-39   229-3#
UDAI1A  229-15#  229-21
UDAI1B  229-17   229-20#
UDAI1C  229-16   229-22#
UDAI1L  229-5#   229-7
UDAI2   229-32#
UDAI2E  229-35   229-40#
UDAI2L  229-34#  229-37
UDAI3   229-38   230-3#
UDAID1  231-16*  233-4#   233-20   233-29
UDAID2  231-17*  231-18*  233-6#
UDAID3  233-8#
```

CROSS REFERENCE TABLE (CREF V04.00)

```
UDAIDT 232-4     233-3#
UDAIEX 229-12    229-19   229-25   229-41   230-22#
UDAINT 212-41    218-13   228-23#  273-3    274-3
UDAIR1 233-3     233-14#
UDAIR2 233-5     233-20#
UDAIR3 233-7     233-29#
UDAIR4 233-9     233-37#
UDAIRC 233-15    233-24   233-32   233-48#
UDAIRX 233-49    233-52#
UDAISE 231-32    232-9    232-11   232-19#
UDAISG 231-29    232-3#
UDAISL 232-8#    232-16
UDAIST 229-11    231-13#
UDAISX 232-14    232-18#
UDARS1 234-25#   234-29   234-32   234-34
UDARS2 234-26    235-3#
UDARSD 144-78*   144-79   232-3*   232-12*  232-13   234-19*  234-25   235-16#  268-8*   268-12*  269-9*   270-26*
UDARSE 234-37    235-6#
UDARSP 229-24    232-8    234-18#  268-9    268-13   269-10   270-27
UDARSX 235-4     235-11#
UDASRV 226-18#   257-23
UF.576 129-12#
UF.CMR 129-3#
UF.CMW 129-4#
UF.INA 129-6#
UF.RPL 129-5#
UF.SCH 129-7#
UF.SCL 129-8#
UF.WBN 129-9#
UF.WPH 129-10#
UF.WPS 129-11#
URNING 136-34#   151-16*  152-39*  152-56*  152-60   155-16*  155-34*  155-41   157-22*  279-29
URUN   136-33#   151-18*  151-20   152-9    152-24   152-46   155-15*  155-20   156-7
UTOT1  181-21#   181-33
UTOT1A 181-24    181-32#
UTOT2  181-22    181-34#
UTOT3  181-39#   181-41
UTOT4  181-36    181-42#
UTOTST 160-13    181-18#
WAITMS 213-16    215-23   218-19   222-11#
WCHNG  269-14    269-22   276-5#
WCHNGD 269-12*   269-20*  276-9    276-24#
X$ALWA 115-12#
X$FALS 115-12#
X$OFFS 115-12#
X$TRUE 115-12#
X1     142-5#    144-15
X13    142-45#   144-59
X1A    142-1#    144-15
X2     142-6#    144-19
X20    142-47#   144-63
X21    142-52#   144-74
X22    142-54#   144-79
X23A   142-56#   144-84
X23B   142-60#   144-88
X24    142-61#   144-98
```

CROSS REFERENCE TABLE (CREF V04.00)

X25	142-63#	144-102		
X26	142-66#	144-107		
X27	142-69#	144-111		
X28	142-71#	144-115		
X29	142-72#	144-119		
X2A	142-2#	144-19		
X3	142-7#	144-23		
X30	142-77#	144-124		
X31	142-78#	144-128		
X32	142-80#	144-132		
X35	142-81#	144-145		
X36	142-82#	144-150		
X37	142-84#	144-154		
X38	142-48#	144-158		
X3A	142-3#	144-23		
X4	142-8#	144-27		
X5	142-13#	144-32		
X6	142-10#	144-45		
X7	142-14#	144-36		
X8	142-11#	144-49		
X8A	142-4#	144-49		
X9	142-15#	144-40		
XFRU	143-8#	144-80	144-94	199-5
XMSG1	143-1#	144-174		
XMSG2	143-2#	144-178		
XPKT1	143-3#	144-161		
XPKT2	143-6#	144-167		
XSA	143-7#	200-5		

CROSS REFERENCE TABLE (CREF V04.00)

```
.BR     116-31#
AND     116-3#  144-69  212-34  218-6   260-27  282-10  282-25
ASSUME  116-39# 117-6   117-10  145-9   155-31  180-24  181-25  248-14  248-28  249-23  256-24  256-25  266-14  269-7
        270-7   270-18  280-57
BCOMPL  167-34  253-14  255-14  255-16  262-21
BGNAUT  264-10
BGNCLN  265-8
BGNHRD  283-14
BGNHW   119-10
BGNINI  252-8
BGNMOD  115-33  121-3   247-3   266-3   283-3
BGNMSG  144-14  144-18  144-22  144-26  144-31  144-35  144-39  144-44  144-48  144-58  144-62  144-66  144-77  144-83
        144-97  144-101 144-106 144-110 144-114 144-118 144-123 144-127 144-131 144-136 144-140 144-144 144-149 144-153
        144-157 145-1
BGNPRO  251-8
BGNPTA  289-3
BGNRPT  247-10
BGNSET  289-1
BGNSFT  285-12
BGNSRV  225-10  226-18  244-5   244-11
BGNSUB  267-1   268-1   269-1   270-1   273-1   274-1   275-1
BGNSW   120-10
BGNTST  266-6   277-3   278-3   279-3
BNCOMP  253-2   253-6   253-10  254-22  256-16  259-6   262-5   262-23  280-16
BREAK   156-43  222-26  231-13  234-27  239-31  245-12  270-35  276-11  282-45
BUILD   191-28# 192-9   192-21
CKLOOP  267-11
CLOCK   255-13  255-15
CLOSE   238-13
CLRVEC  231-27  267-7   272-10
DESCRI  139-17
DEVTYP  139-12
DISPAT  118-9
DISPLA  286-4
DOCLN   147-8   150-39  243-7   245-30  254-30  262-9   263-4   263-9   263-15  263-21
DORPT   151-32  157-10  245-29  253-12  282-48
END     122-29# 132-47  134-24  283-27  285-29
ENDAUT  264-12
ENDCLN  265-21
ENDHRD  284-9
ENDHW   119-18
ENDINI  263-23
ENDMOD  120-21  246-38  265-23  282-53  288-11
ENDMSG  144-16  144-20  144-24  144-28  144-33  144-37  144-41  144-46  144-50  144-60  144-64  144-75  144-81  144-95
        144-99  144-103 144-108 144-112 144-116 144-120 144-125 144-129 144-134 144-138 144-142 144-147 144-151 144-155
        144-159 145-22
ENDPRO  251-14
ENDPTA  289-12
ENDRPT  250-43
ENDSET  289-14
ENDSFT  286-15
ENDSRV  225-14  226-21  244-9   244-13
ENDSUB  267-14  268-15  269-32  272-11  273-4   274-6   275-10
ENDSW   120-19
ENDTST  275-14  277-24  278-14  282-51
ENTRY   192-3#  192-9   192-9   192-9   192-9   192-9   192-9   192-9   192-9   192-15# 192-21  192-21  192-21  192-21
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

```
M$EXSE  250-12#  262-31#  266-18#  282-49#
M$EXTJ  250-12   250-12#  262-31#  266-18#  282-49#
M$GEN   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  118-9    118-9#   119-10   119-10   119-10#  119-10#  119-18   119-18#
        120-10   120-10   120-10#  120-10#  120-19   120-19#  136-8    136-8#   139-12   139-12#  139-17   139-17#  144-14   144-14#
        144-16   144-16#  144-18   144-18#  144-20   144-20#  144-22   144-22#  144-24   144-24#  144-26   144-26#  144-28   144-28#
        144-31   144-31#  144-33   144-33#  144-35   144-35#  144-37   144-37#  144-39   144-39#  144-41   144-41#  144-44   144-44#
        144-46   144-46#  144-48   144-48#  144-50   144-50#  144-58   144-58#  144-60   144-60#  144-62   144-62#  144-64   144-64#
        144-66   144-66#  144-75   144-75#  144-77   144-77#  144-81   144-81#  144-83   144-83#  144-95   144-95#  144-97   144-97#
        144-99   144-99#  144-101  144-101#  144-103  144-103#  144-106  144-106#  144-108  144-108#  144-110  144-110#  144-112  144-112#
        144-114  144-114#  144-116  144-116#  144-118  144-118#  144-120  144-120#  144-123  144-123#  144-125  144-125#  144-127  144-127#
        144-129  144-129#  144-131  144-131#  144-134  144-134#  144-136  144-136#  144-138  144-138#  144-140  144-140#  144-142  144-142#
        144-144  144-144#  144-147  144-147#  144-149  144-149#  144-151  144-151#  144-153  144-153#  144-155  144-155#  144-157  144-157#
        144-159  144-159#  145-1    145-1#   145-22   145-22#  170-3    170-3#   225-10   225-10#  225-14   225-14#  226-18   226-18#
        226-21   226-21#  244-5    244-5#   244-9    244-9#   244-11   244-11#  244-13   244-13#  247-10   247-10#  250-43   250-43#
        251-8    251-8#   252-8    252-8#   262-6    262-6#   263-23   263-23#  264-10   264-10#  264-12   264-12#  265-8    265-8#
        265-21   265-21#  266-6    266-6#   267-1    267-1#   267-14   267-14#  268-1    268-1#   268-15   268-15#  269-1    269-1#
        269-32   269-32#  270-1    270-1#   272-11   272-11#  273-1    273-1#   273-4    273-4#   274-1    274-1#   274-6    274-6#
        275-1    275-1#   275-10   275-10#  275-14   275-14#  277-3    277-3#   277-24   277-24#  278-3    278-3#   278-14   278-14#
        279-3    279-3#   282-51   282-51#  283-14   283-14#  284-9    284-9#   285-12   285-12#  286-15   286-15#  288-9    288-9#
        289-3    289-3#   289-12   289-12#
M$GENB  170-3    170-3#   262-6    262-6#
M$GETS  119-18   119-18#  120-19   120-19#  120-21   120-21#  144-16   144-16#  144-20   144-20#  144-24   144-24#  144-28   144-28#
        144-33   144-33#  144-37   144-37#  144-41   144-41#  144-46   144-46#  144-50   144-50#  144-60   144-60#  144-64   144-64#
        144-75   144-75#  144-81   144-81#  144-95   144-95#  144-99   144-99#  144-103  144-103#  144-108  144-108#  144-112  144-112#
        144-116  144-116#  144-120  144-120#  144-125  144-125#  144-129  144-129#  144-134  144-134#  144-138  144-138#  144-142  144-142#
        144-147  144-147#  144-151  144-151#  144-155  144-155#  144-159  144-159#  145-22   145-22#  225-14   225-14#  226-21   226-21#
        244-9    244-9#   244-13   244-13#  246-38   246-38#  250-43   250-43#  251-14   251-14#  263-23   263-23#  264-12   264-12#
        265-21   265-21#  265-23   265-23#  267-14   267-14#  268-15   268-15#  269-32   269-32#  272-11   272-11#  273-4    273-4#
        274-6    274-6#   275-10   275-10#  275-14   275-14#  277-24   277-24#  278-14   278-14#  282-51   282-51#  282-53   282-53#
        284-9    284-9#   286-15   286-15#  288-11   288-11#
M$GETT  250-12#  262-31#  266-18#  282-49#
M$GNGB  115-33#  115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  118-9    118-9#   119-10   119-10   119-10#  120-10   120-10
        120-10#  121-3#   136-8    136-8#   139-12   139-12#  139-17   139-17#  144-14   144-14#  144-18   144-18#  144-22   144-22#
        144-26   144-26#  144-31   144-31#  144-35   144-35#  144-39   144-39#  144-44   144-44#  144-48   144-48#  144-58   144-58#
        144-62   144-62#  144-66   144-66#  144-77   144-77#  144-83   144-83#  144-97   144-97#  144-101  144-101#  144-106  144-106#
        144-110  144-110#  144-114  144-114#  144-118  144-118#  144-123  144-123#  144-127  144-127#  144-131  144-131#  144-136  144-136#
        144-140  144-140#  144-144  144-144#  144-149  144-149#  144-153  144-153#  144-157  144-157#  145-1    145-1#   225-10   225-10#
        226-18   226-18#   244-5    244-5#   244-11   244-11#  247-3    247-10   247-10#  251-8    251-8#   252-8    252-8#   264-10
        264-10#  265-8    265-8#   266-3#   283-3#   283-14   283-14#  285-12   285-12#  288-9    288-9#
M$GNIN  115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42   115-42
        115-42   115-42   115-42   115-42   115-42   115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  115-42#
        115-42#  115-42#  115-42#  115-42#  115-42#  115-42#  118-9    118-9    118-9    118-9    118-9    118-9#   118-9#   118-9#
```

CROSS REFERENCE TABLE (CREF V04.00)

	272-10	272-10	272-10#	272-10#	272-11	272-11#	273-1	273-1#	273-4	273-4#	274-1	274-1#	274-6	274-6#
	275-1	275-1#	275-10	275-10#	275-14	275-14#	276-11	276-11#	276-19	276-19	276-19	276-19	276-19#	276-19#
	276-19#	276-19#	276-19#	277-24	277-24#	278-14	278-14#	280-15	280-15#	280-16	280-16#	282-45	282-45#	282-48
	282-48#	282-49	282-49	282-49#	282-49#	282-51	282-51#	283-14	283-14#	284-1	284-1	284-1	284-1	284-1#
	284-2	284-2	284-2	284-2	284-2#	284-3	284-3	284-3	284-3	284-3	284-3#	284-4	284-4	284-4
	284-4	284-4	284-4#	284-5	284-5	284-5	284-5	284-5	284-5#	284-7	284-7	284-7	284-7#	284-9
	284-9#	285-12	285-12#	286-1	286-1	286-1	286-1#	286-4	286-4	286-4#	286-4#	286-6	286-6	286-6
	286-6	286-6	286-6#	286-7	286-7	286-7	286-7	286-7	286-7#	286-8	286-8	286-8	286-8#	286-9
	286-9	286-9	286-9#	286-10	286-10	286-10	286-10#	286-15	286-15#	288-9	288-9	288-9	288-9#	289-3
	289-3	289-3#	289-3#											
M$GNLS	170-3	170-3#	262-6	262-6#										
M$GNSU	267-1	267-1#	268-1	268-1#	269-1	269-1#	270-1	270-1#	273-1	273-1#	274-1	274-1#	275-1	275-1#
M$GNTA	119-18	119-18#	120-19	120-19#	144-16	144-16#	144-20	144-20#	144-24	144-24#	144-28	144-28#	144-33	144-33#
	144-37	144-37#	144-41	144-41#	144-46	144-46#	144-50	144-50#	144-60	144-60#	144-64	144-64#	144-75	144-[illegible]#
	144-81	144-81#	144-95	144-95#	144-99	144-99#	144-103	144-103#	144-108	144-108#	144-112	144-112#	144-116	144-116#
	144-120	144-120#	144-125	144-125#	144-129	144-129#	144-134	144-134#	144-138	144-138#	144-142	144-142#	144-147	144-147#
	144-151	144-151#	144-155	144-155#	144-159	144-159#	145-22	145-22#	225-14	225-14#	226-21	226-21#	244-9	244-9#
	244-13	244-13#	250-43	250-43#	263-23	263-23#	264-12	264-12#	265-21	265-21#	267-14	267-14#	268-15	268-15#
	269-32	269-32#	272-11	272-11#	273-4	273-4#	274-6	274-6#	275-10	275-10#	275-14	275-14#	277-24	277-24#
	278-14	278-14#	282-51	282-51#	284-9	284-9#	286-15	286-15#	289-3	289-3#	289-12	289-12#		
M$GNTE	266-6	266-6#	277-3	277-3#	278-3	278-3#	279-3	279-3#						
M$HAPT	115-42	115-42#												
M$HNAP	115-42	115-42#												
M$INCR	115-33	115-33#	119-10	119-10	119-10#	119-10#	120-10	120-10	120-10#	120-10#	121-3	121-3#	144-14	144-14
	144-14#	144-14#	144-16#	144-18	144-18	144-18#	144-18#	144-20#	144-22	144-22	144-22#	144-22#	144-24#	144-26
	144-26	144-26#	144-26#	144-28#	144-31	144-31	144-31#	144-31#	144-33#	144-35	144-35	144-35#	144-35#	144-37#
	144-39	144-39	144-39#	144-39#	144-41#	144-44	144-44	144-44#	144-44#	144-46#	144-48	144-48	144-48#	144-48#
	144-50#	144-58	144-58	144-58#	144-58#	144-60#	144-62	144-62	144-62#	144-62#	144-64#	144-66	144-66	144-66#
	144-66#	144-75#	144-77	144-77	144-77#	144-77#	144-80#	144-81#	144-83	144-83	144-83#	144-83#	144-95#	144-97
	144-97	144-97#	144-97#	144-99#	144-101	144-101	144-101#	144-101#	144-103#	144-106	144-106	144-106#	144-106#	144-108#
	144-110	144-110	144-110#	144-110#	144-112#	144-114	144-114	144-114#	144-114#	144-116#	144-118	144-118	144-118#	144-118#
	144-120#	144-123	144-123	144-123#	144-123#	144-125#	144-127	144-127	144-127#	144-127#	144-129#	144-131	144-131	144-131#
	144-131#	144-134#	144-136	144-136	144-136#	144-136#	144-138#	144-140	144-140	144-140#	144-140#	144-142#	144-144	144-144
	144-144#	144-144#	144-147#	144-149	144-149	144-149#	144-149#	144-151#	144-153	144-153	144-153#	144-153#	144-155#	144-157
	144-157	144-157#	144-157#	144-159#	145-1	145-1	145-1#	145-1#	145-22#	147-7#	147-8#	150-38#	150-39#	151-31#
	151-32#	153-1#	156-22#	156-40#	156-43#	157-10#	158-21#	159-11#	167-33#	170-3	170-3#	170-3#	180-49#	182-37#
	184-44#	202-24#	209-13#	209-15#	209-17#	209-19#	211-41#	212-37#	216-3#	218-9#	222-26#	222-34#	223-5#	225-10
	225-10	225-10#	225-10#	226-18	226-18	226-18#	226-18#	229-18#	229-40#	231-13#	231-24#	231-27#	231-30#	233-50#
	234-27#	234-36#	235-5#	238-13#	239-28#	239-31#	239-33#	239-38#	240-7#	241-17#	241-24#	242-1#	242-12#	242-15#
	243-6#	243-7#	244-5	244-5	244-5#	244-5#	244-11	244-11	244-11#	244-11#	245-11#	245-12#	245-27#	245-29#
	245-30#	247-3	247-3#	247-10	247-10	247-10#	247-10#	249-22#	249-24#	250-43#	251-8	251-8	251-8#	251-8#
	252-8	252-8	252-8#	252-8#	253-1#	253-5#	253-9#	253-12#	253-13#	254-21#	254-29#	254-30#	255-2#	255-13#
	255-15#	255-24#	256-15#	259-5#	261-36#	262-4#	262-6	262-6#	262-6#	262-9#	262-17#	262-20#	262-22#	262-31#
	263-3#	263-4#	263-8#	263-9#	263-14#	263-15#	263-20#	263-21#	263-23#	264-10	264-10	264-10#	264-10#	264-12#
	265-8	265-8	265-8#	265-8#	265-21#	266-3	266-3#	266-6	266-6	266-6	266-6#	266-6#	266-6#	266-18#
	267-1	267-1	267-1	267-1#	267-1#	267-1#	267-3#	267-7#	267-10#	267-11#	267-14#	268-1	268-1	268-1
	268-1#	268-1#	268-1#	268-15#	269-1	269-1	269-1	269-1#	269-1#	269-1#	269-26#	269-32#	270-1	270-1
	270-1	270-1#	270-1#	270-1#	270-19#	270-20#	270-35#	271-10#	271-21#	271-27#	272-1#	272-5#	272-10#	272-11#
	273-1	273-1	273-1	273-1#	273-1#	273-1#	273-4#	274-1	274-1	274-1	274-1#	274-1#	274-1#	274-6#
	275-1	275-1	275-1	275-1#	275-1#	275-1#	275-10#	275-14#	276-11#	276-19#	277-3	277-3	277-3	277-3#
	277-3#	277-3#	277-24#	278-3	278-3	278-3	278-3#	278-3#	278-3#	278-14#	279-3	279-3	279-3	279-3#
	279-3#	279-3#	280-15#	282-45#	282-48#	282-49#	282-51#	283-3	283-3#	283-14	283-14	283-14#	283-14#	285-12
	285-12	285-12#	285-12#	289-1	289-1#	289-3	289-3	289-3	289-3#					
M$LDRO	231-27	231-27#	239-28	239-28#	253-1	253-1#	253-5	253-5#	253-9	253-9#	253-13	253-13#	254-21	254-21#
	255-13	255-13#	255-15	255-15#	256-15	256-15#	259-5	259-5#	262-17	262-17#	262-20	262-20#	262-22	262-22#
	267-7	267-7#	270-20	270-20#	271-10	271-10#	271-21	271-21#	272-1	272-1#	272-10	272-10#		

CROSS REFERENCE TABLE (CREF V04.00)

```
M$MCHI  115-12   115-12#
M$MCLO  115-12   115-12#
M$POP   119-18   119-18#  120-19   120-19#  120-21   120-21#  144-16   144-16#  144-20   144-20#  144-24   144-24#  144-28   144-28#
        144-33   144-33#  144-37   144-37#  144-41   144-41#  144-46   144-46#  144-50   144-50#  144-60   144-60#  144-64   144-64#
        144-75   144-75#  144-81   144-81#  144-95   144-95#  144-99   144-99#  144-103  144-103# 144-108  144-108# 144-112  144-112#
        144-116  144-116# 144-120  144-120# 144-125  144-125# 144-129  144-129# 144-134  144-134# 144-138  144-138# 144-142  144-142#
        144-147  144-147# 144-151  144-151# 144-155  144-155# 144-159  144-159# 145-22   145-22#  225-14   225-14#  226-21   226-21#
        244-9    244-9#   244-13   244-13#  246-38   246-38#  250-43   250-43#  251-14   251-14#  263-23   263-23#  264-12   264-12#
        265-21   265-21#  265-23   265-23#  267-14   267-14#  268-15   268-15#  269-32   269-32#  272-11   272-11#  273-4    273-4#
        274-6    274-6#   275-10   275-10#  275-14   275-14#  277-24   277-24#  278-14   278-14#  282-51   282-51#  282-53   282-53#
        284-9    284-9#   286-15   286-15#  288-11   288-11#
M$PRIN  144-80   144-80#  209-13   209-13#  209-15   209-15#  209-17   209-17#  209-19   209-19#  249-22   249-22#  249-24   249-24#
        261-36   261-36#
M$PUSH  115-33   115-33#  119-10   119-10#  120-10   120-10#  121-3    121-3#   144-14   144-14#  144-18   144-18#  144-22   144-22#
        144-26   144-26#  144-31   144-31#  144-35   144-35#  144-39   144-39#  144-44   144-44#  144-48   144-48#  144-58   144-58#
        144-62   144-62#  144-66   144-66#  144-77   144-77#  144-83   144-83#  144-97   144-97#  144-101  144-101# 144-106  144-106#
        144-110  144-110# 144-114  144-114# 144-118  144-118# 144-123  144-123# 144-127  144-127# 144-131  144-131# 144-136  144-136#
        144-140  144-140# 144-144  144-144# 144-149  144-149# 144-153  144-153# 144-157  144-157# 145-1    145-1#   225-10   225-10#
        226-18   226-18#  244-5    244-5#   244-11   244-11#  247-3    247-3#   247-10   247-10#  251-8    251-8#   252-8    252-8#
        264-10   264-10#  265-8    265-8#   266-3    266-3#   266-6    266-6#   267-1    267-1#   268-1    268-1#   269-1    269-1#
        270-1    270-1#   273-1    273-1#   274-1    274-1#   275-1    275-1#   277-3    277-3#   278-3    278-3#   279-3    279-3#
        283-3    283-3#   283-14   283-14#  285-12   285-12#
M$PUT   144-80   144-80   144-80#  209-13   209-13   209-13   209-13#  209-15   209-15   209-15   209-15#  209-17   209-17   209-17
        209-17#  209-19   209-19   209-19   209-19#  212-37   212-37   212-37   212-37   212-37#  218-9    218-9    218-9    218-9
        218-9#   231-24   231-24   231-24   231-24   231-24#  245-11   245-11   245-11   245-11   245-11#  249-22   249-22   249-22
        249-22   249-22   249-22   249-22   249-22   249-22#  249-24   249-24   249-24   249-24   249-24   249-24#  255-24   255-24
        255-24   255-24   255-24#  261-36   261-36   261-36   261-36   261-36   261-36#  267-3    267-3    267-3    267-3    267-3#
        270-19   270-19   270-19   270-19   270-19#
M$PUT1  144-80   144-80   144-80#  144-80#  209-13   209-13   209-13   209-13#  209-13#  209-13#  209-15   209-15   209-15   209-15#
        209-15#  209-15#  209-17   209-17   209-17   209-17#  209-17#  209-17#  209-19   209-19   209-19   209-19#  209-19#  209-19#
        212-37   212-37   212-37   212-37   212-37#  212-37#  212-37#  212-37#  218-9    218-9    218-9    218-9    218-9#   218-9#
        218-9#   218-9#   231-24   231-24   231-24   231-24   231-24#  231-24#  231-24#  231-24#  245-11   245-11   245-11   245-11
        245-11#  245-11#  245-11#  245-11#  249-22   249-22   249-22   249-22   249-22   249-22   249-22   249-22   249-22#  249-22#
        249-22#  249-22#  249-22#  249-22#  249-22#  249-22#  249-24   249-24   249-24   249-24   249-24   249-24#  249-24#  249-24#
        249-24#  249-24#  255-24   255-24   255-24   255-24   255-24#  255-24#  255-24#  255-24#  261-36   261-36   261-36   261-36
        261-36   261-36#  261-36#  261-36#  261-36#  261-36#  267-3    267-3    267-3    267-3    267-3#   267-3#   267-3#   267-3#
        270-19   270-19   270-19   270-19   270-19#  270-19#  270-19#  270-19#
M$RADI  170-3    170-3#   262-6    262-6#   284-1    284-1#   284-2    284-2#   284-3    284-3#   284-4    284-4#   284-5    284-5#
        284-7    284-7#   286-1    286-1#   286-6    286-6#   286-7    286-7#   286-8    286-8#   286-9    286-9#   286-10   286-10#
M$RBRO  239-33   239-33#  239-38   239-38#  240-7    240-7#   241-17   241-17#  241-24   241-24#  242-1    242-1#   242-12   242-12#
        242-15   242-15#
M$RNRO  180-49   180-49#  182-37   182-37#  184-44   184-44#  254-21   254-21#  255-2    255-2#   255-13   255-13#  255-15   255-15#
        256-15   256-15#  259-5    259-5#
M$SETS  115-33   115-33#  119-10   119-10#  120-10   120-10#  121-3    121-3#   144-14   144-14#  144-18   144-18#  144-22   144-22#
        144-26   144-26#  144-31   144-31#  144-35   144-35#  144-39   144-39#  144-44   144-44#  144-48   144-48#  144-58   144-58#
        144-62   144-62#  144-66   144-66#  144-77   144-77#  144-83   144-83#  144-97   144-97#  144-101  144-101# 144-106  144-106#
        144-110  144-110# 144-114  144-114# 144-118  144-118# 144-123  144-123# 144-127  144-127# 144-131  144-131# 144-136  144-136#
        144-140  144-140# 144-144  144-144# 144-149  144-149# 144-153  144-153# 144-157  144-157# 145-1    145-1#   225-10   225-10#
        226-18   226-18#  244-5    244-5#   244-11   244-11#  247-3    247-3#   247-10   247-10#  251-8    251-8#   252-8    252-8#
        264-10   264-10#  265-8    265-8#   266-3    266-3#   266-6    266-6#   267-1    267-1#   268-1    268-1#   269-1    269-1#
        270-1    270-1#   273-1    273-1#   274-1    274-1#   275-1    275-1#   277-3    277-3#   278-3    278-3#   279-3    279-3#
        283-3    283-3#   283-14   283-14#  285-12   285-12#
M$SVC   144-16   144-16#  144-20   144-20#  144-24   144-24#  144-28   144-28#  144-33   144-33#  144-37   144-37#  144-41   144-41#
        144-46   144-46#  144-50   144-50#  144-60   144-60#  144-64   144-64#  144-75   144-75#  144-80   144-80#  144-81   144-81#
        144-95   144-95#  144-99   144-99#  144-103  144-103# 144-108  144-108# 144-112  144-112# 144-116  144-116# 144-120  144-120#
        144-125  144-125# 144-129  144-129# 144-134  144-134# 144-138  144-138# 144-142  144-142# 144-147  144-147# 144-151  144-151#
```

```
        144-155  144-155#  144-159  144-159#  145-22   145-22#  147-7    147-8    147-8#   150-38   150-39   150-39#  151-31   151-32
        151-32#  153-1    156-22   156-40    156-43   156-43#  157-10   157-10#  158-21   159-11   167-33   167-33#  170-3    170-3#
        180-49   180-49#  182-37   182-37#   184-44   184-44#  202-24   209-13   209-13#  209-15   209-15#  209-17   209-17#  209-19
        209-19#  211-41   211-41#  212-37    212-37#  216-3    218-9    218-9#   222-26   222-26#  222-34   223-5    229-18   229-40
        231-13   231-13#  231-24   231-24#   231-27   231-27#  231-30   233-50   234-27   234-27#  234-36   235-5    238-13   238-13#
        239-28   239-28#  239-31   239-31#   239-33   239-33#  239-38   239-38#  240-7    240-7#   241-17   241-17#  241-24   241-24#
        242-1    242-1#   242-12   242-12#   242-15   242-15#  243-6    243-7    243-7#   245-11   245-11#  245-12   245-12#  245-27
        245-29   245-29#  245-30   245-30#   249-22   249-22#  249-24   249-24#  250-12#  250-43   250-43#  253-1    253-1#   253-5
        253-5#   253-9    253-9#   253-12    253-12#  253-13   253-13#  254-21   254-21#  254-29   254-30   254-30#  255-2    255-2#
        255-13   255-13#  255-15   255-15#   255-24   255-24#  256-15   256-15#  259-5    259-5#   261-36   261-36#  262-4    262-4#
        262-6    262-6#   262-9    262-9#    262-17   262-17#  262-20   262-20#  262-22   262-22#  262-31   262-31#  263-3    263-4
        263-4#   263-8    263-9    263-9#    263-14   263-15   263-15#  263-20   263-21   263-21#  263-23   263-23#  264-12   264-12#
        265-21   265-21#  266-18   266-18#   267-1    267-1#   267-3    267-3#   267-7    267-7#   267-10   267-11   267-11#  267-14
        267-14#  268-1    268-1#   268-15    268-15#  269-1    269-1#   269-26   269-32   269-32#  270-1    270-1#   270-19   270-19#
        270-20   270-20#  270-35   270-35#   271-10   271-10#  271-21   271-21#  271-27   272-1    272-1#   272-5    272-10   272-10#
        272-11   272-11#  273-1    273-1#    273-4    273-4#   274-1    274-1#   274-6    274-6#   275-1    275-1#   275-10   275-10#
        275-14   275-14#  276-11   276-11#   276-19   277-24   277-24#  278-14   278-14#  280-15   280-15#  282-45   282-45#  282-48
        282-48#  282-49   282-49#  282-51    282-51#
MSTLAB  144-16#  144-20#  144-24#  144-28#   144-33#  144-37#  144-41#  144-46#  144-50#  144-60#  144-64#  144-75#  144-80#  144-81#
        144-95#  144-99#  144-103#  144-108#  144-112#  144-116#  144-120#  144-125#  144-129#  144-134#  144-138#  144-142#  144-147#  144-151#
        144-155#  144-159#  145-22#  147-7#   147-8#   150-38#  150-39#  151-31#  151-32#  153-1#   156-22#  156-40#  156-43#  157-10#
        158-21#  159-11#  167-33#  170-3#   180-49#  182-37#  184-44#  202-24#  209-13#  209-15#  209-17#  209-19#  211-41#  212-37#
        216-3#   218-9#   222-26#  222-34#  223-5#   229-18#  229-40#  231-13#  231-24#  231-27#  231-30#  233-50#  234-27#  234-36#
        235-5#   238-13#  239-28#  239-31#  239-33#  239-38#  240-7#   241-17#  241-24#  242-1#   242-12#  242-15#  243-6#   243-7#
        245-11#  245-12#  245-27#  245-29#  245-30#  249-22#  249-24#  250-43#  253-1#   253-5#   253-9#   253-12#  253-13#  254-21#
        254-29#  254-30#  255-2#   255-13#  255-15#  255-24#  256-15#  259-5#   261-36#  262-4#   262-6#   262-9#   262-17#  262-20#
        262-22#  262-31#  263-3#   263-4#   263-8#   263-9#   263-14#  263-15#  263-20#  263-21#  263-23#  264-12#  265-21#  266-18#
        267-1#   267-3#   267-7#   267-10#  267-11#  267-14#  268-1#   268-15#  269-1#   269-26#  269-32#  270-1#   270-19#  270-20#
        270-35#  271-10#  271-21#  271-27#  272-1#   272-5#   272-10#  272-11#  273-1#   273-4#   274-1#   274-6#   275-1#   275-10#
        275-14#  276-11#  276-19#  277-24#  278-14#  280-15#  282-45#  282-48#  282-49#  282-51#
MSTSTL  144-16   144-16#  144-20   144-20#   144-24   144-24#  144-28   144-28#  144-33   144-33#  144-37   144-37#  144-41   144-41#
        144-46   144-46#  144-50   144-50#   144-60   144-60#  144-64   144-64#  144-75   144-75#  144-80   144-80#  144-81   144-81#
        144-95   144-95#  144-99   144-99#   144-103  144-103# 144-108  144-108# 144-112  144-112# 144-116  144-116# 144-120  144-120#
        144-125  144-125# 144-129  144-129#  144-134  144-134# 144-138  144-138# 144-142  144-142# 144-147  144-147# 144-151  144-151#
        144-155  144-155# 144-159  144-159#  145-22   145-22#  147-7    147-7#   147-7#   147-8    147-8#   150-38   150-38#  150-38#
        150-39   150-39#  151-31   151-31#   151-31#  151-32   151-32#  153-1    153-1#   153-1#   156-22   156-22#  156-22#  156-40
        156-40#  156-40#  156-43   156-43#   157-10   157-10#  158-21   158-21#  158-21#  159-11   159-11#  159-11#  167-33   167-33#
        170-3    170-3#   180-49   180-49#   182-37   182-37#  184-44   184-44#  202-24   202-24#  202-24#  209-13   209-13#  209-15
        209-15#  209-17   209-17#  209-19    209-19#  211-41   211-41#  212-37   212-37#  216-3    216-3#   216-3#   218-9    218-9#
        222-26   222-26#  222-34   222-34#   222-34#  223-5    223-5#   223-5#   229-18   229-18#  229-18#  229-40   229-40#  229-40#
        231-13   231-13#  231-24   231-24#   231-27   231-27#  231-30   231-30#  231-30#  233-50   233-50#  233-50#  234-27   234-27#
        234-36   234-36#  234-36#  235-5     235-5#   235-5#   238-13   238-13#  239-28   239-28#  239-31   239-31#  239-33   239-33#
        239-38   239-38#  240-7    240-7#    241-17   241-17#  241-24   241-24#  242-1    242-1#   242-12   242-12#  242-15   242-15#
        243-6    243-6#   243-6#   243-7     243-7#   245-11   245-11#  245-12   245-12#  245-27   245-27#  245-27#  245-29   245-29#
        245-30   245-30#  249-22   249-22#   249-24   249-24#  250-43   250-43#  253-1    253-1#   253-5    253-5#   253-9    253-9#
        253-12   253-12#  253-13   253-13#   254-21   254-21#  254-29   254-29#  254-29#  254-30   254-30#  255-2    255-2#   255-13
        255-13#  255-15   255-15#  255-24    255-24#  256-15   256-15#  259-5    259-5#   261-36   261-36#  262-4    262-4#   262-6
        262-6#   262-9    262-9#   262-17    262-17#  262-20   262-20#  262-22   262-22#  262-31   262-31#  263-3    263-3#   263-3#
        263-4    263-4#   263-8    263-8#    263-8#   263-9    263-9#   263-14   263-14#  263-14#  263-15   263-15#  263-20   263-20#
        263-20#  263-21   263-21#  263-23    263-23#  264-12   264-12#  265-21   265-21#  266-18   266-18#  267-1    267-1#   267-3
        267-3#   267-7    267-7#   267-10    267-10#  267-10#  267-11   267-11#  267-14   267-14#  268-1    268-1#   268-15   268-15#
        269-1    269-1#   269-26   269-26#   269-26#  269-32   269-32#  270-1    270-1#   270-19   270-19#  270-20   270-20#  270-35
        270-35#  271-10   271-10#  271-21    271-21#  271-27   271-27#  271-27#  272-1    272-1#   272-5    272-5#   272-5#   272-10
        272-10#  272-11   272-11#  273-1     273-1#   273-4    273-4#   274-1    274-1#   274-6    274-6#   275-1    275-1#   275-10
        275-10#  275-14   275-14#  276-11    276-11#  276-19   276-19#  276-19#  277-24   277-24#  278-14   278-14#  280-15   280-15#
        282-45   282-45#  282-48   282-48#   282-49   282-49#  282-51   282-51#
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

A 151-34
U 151-34