RA60/80/81 UDA50/KDA50-Q DRV EXE
UDA50A KDA50Q CZUDIAO

AH-T931A-MC FICHE
FICHE 01 OF 02

FEB 1985
digital
Made In USA

RA60/80/81 UDA50/KDA50-Q DRV EXE
UDA50A KDA50Q CZUDIA0

AH-T931A-MC FICHE
FICHE 02 OF 02

FEB 1985
digital
Made In USA

IDENTIFICATION

PRODUCT CODE: AC-T930A-MC

PRODUCT NAME: CZUDIAO UDA50A/KDA50Q DRIVE EXERCISER

PRODUCT DATE: 7-OCT-1984

MAINTAINER: ROGER OAKEY

AUTHOR: JOHN MERTZ

```
1
2
3
        .REM @
```

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

This diagnostic program is provided for testing the UDA50-A or KDA50-Q Disk Controller and the associated disk drives. There is one test within this diagnostic:

Test # 1 - 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 UDA50-A or KDA50-Q without modifying or rereleasing the program. 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.

If this program is run on a system that does not support memory management, the program is limited to testing one controller and one drive at a time.

The following PDP-11 diagnostic programs are also provided for the UDA50-A or KDA50-Q disk subsystem:

CZUDHA0 - UDA50-A/KDA50-Q Basic Subsystem Diagnostic.

CZUDKA0 - UDA50-A/KDA50-Q Formatter.

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.

This diagnostic can isolate many controller faults to either the processor module or the SDI module of the controller. Whenever a controller fault is detected and the fault can be isolated to one of the two controller modules, that module should be replaced.

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
- One or more UDA50-A or KDA50-Q subsystems. The subsystem controller must be at the latest hardware and microcode revision level.
- Line clock - either Type L or P

The line clock is used for all timed loops in the program. If a clock exists on the system it should be enabled so it can interrupt the processor. The diagnostic will run on a system with no clock or with the clock disabled, 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).

The XXDP+ system device does not need to 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.

CSR ADDRESS OF CONTROLLER (O) 172150 ?

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

DRIVE # (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. A maximum of four disk drives may be tested on one controller at a time (controller configuration limit).

EXERCISE ON CUSTOMER DATA AREA (L) N ?

Answer "N" to have this test 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     CONTROLLER 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 Runtime 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 ?

This test has a manual intervention mode which allows 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, the test 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.

ERROR LIMIT (D) 32 ?

Enter the number of hard errors allowed before a drive is dropped from exercise. 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, 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 this test is to run. Answer with a zero to prevent test from ending. The only other way this test can end is to have all drives dropped because the error limit on each is exceeded. Of course, the operator can always stop the test by typing a control-C.

SUPPRESS PRINTING SOFT ERRORS (L) Y?

When the test 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 this test 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 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 the test is stopped and a statistical report requested. This log feature is intended to allow the Digital Diagnosis Center (DDC) to start this test 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. This test will not be allowed to end while the error log is enabled until the error log is printed. The log buffer will hold 16 error messages. Using the log buffer will decrease the number of units that can be simultaneously 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 are testing multiple drives on the same controller, this becomes tedious since most of the answers are repetitious.

To illustrate a more efficient method, suppose you are testing a UDA50-A with 4 disk drives attached to it. These units are numbered 0 through 3. There is one hardware parameter that can vary among units, the drive number. This drive number may be 0 through 252. Below is a simple way to build a table for one UDA50-A with four units.

```
# UNITS (D) ?  4<CR>

UNIT 0
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ? <CR>
DRIVE # (D) 0 ? 0<CR>
EXERCISE ON CUSTOMER DATA AREA (L) N ? <CR>

UNIT 1
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ? <CR>
DRIVE # (D) 0 ? 1<CR>
EXERCISE ON CUSTOMER DATA AREA (L) N ? <CR>

UNIT 2
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ? <CR>
DRIVE # (D) 1 ? 2<CR>
EXERCISE ON CUSTOMER DATA AREA (L) N ? <CR>

UNIT 3
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ? <CR>
DRIVE # (D) 2 ? 3<CR>
EXERCISE ON CUSTOMER DATA AREA (L) N ? <CR>
```

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. Also, notice that the default value for the drive number changes when a non-default response is given.

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

Example 1:

```
# UNITS (D) ?  4<CR>

UNIT 0
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ? <CR>
DRIVE # (D) 0 ? 0-3<CR>
EXERCISE ON CUSTOMER DATA AREA (L) N ? <CR>
```

Example 2:

```
# UNITS (D) ?  4<CR>

UNIT 0
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ? <CR>
DRIVE # (D) 0 ? 0,1,2,3<CR>
EXERCISE ON CUSTOMER DATA AREA (L) N ? <CR>
```

As you can see in the above dialogue, the supervisor will build as many entries as it can with the information given in any one pass through the questions. In each example, four entries are built since four drive numbers were specified. The supervisor assumes that the controller CSR address is 172150 and the diagnostic should not exercise on the customer data area for each entry since these values were specified only once. In the first example, the "-" construct tells the supervisor to increment the data from the first number to the second. In this case, drive numbers 0, 1, 2, and 3 were specified.

2.7 QUICK START-UP PROCEDURE

To start-up this program:

1. Boot XXDP+
2. Enter the date.
3. On certain systems you will be asked if the system has a Unibus.
   If this question is asked, answer it either "Y" or "N".
4. Type "R ZUDIAO"
5. Type "START"
6. Answer the "CHANGE HW" question with "Y"
7. Answer all the hardware questions
8. 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 UDA50-A or KDA50-Q:

```
DR>STA/FLA:PNT

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
UNIBUS ADDRESS OF CONTROLLER (O)  172150 ?
DRIVE NUMBER (D) 0,1
EXERCISE ON CUSTOMER DATA AREA (L) N ?

CHANGE SW (L)  ? N

TST: 001

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

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

TEST 1 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/PAS:1

CHANGE HW (L)  ? N

CHANGE SW (L)  ? Y

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

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

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

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

TEST 1 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

CZUDC EOP     1
    0 CUMULATIVE ERRORS
DR>
```

F2

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:

```
diag severity errnum ON UNIT unit TST tst SUB sub PC: hostpc
error message
```

where:
- diag = diagnostic name
- severity = error type (SYS FTL ERR, DEV FTL ERR, HRD ERR or SFT ERR)
- errnum = error number
- unit = Arbitrary number assigned by the supervisor to each P-table
- tst = test where error occurred
- sub = subtst where error occurred
- hostpc = address of error message call in the host program

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

Device fatal errors (DVC FTL ERR) are used to report errors that are fatal to the device (may be either a UDA50-A or KDA50-Q 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 not used by this diagnostic program.

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 ZUDIAO" command to the XXDP+ monitor contains two parts, the host level code and a program which is loaded into the UDA50-A or KDA50-Q for execution. The controller program is called a "diagnostic machine" or DM program. The "diagnostic machine" is the facility in the controller which executes a PDP-11 like pogram. The large majority of the testing is done by the "diagnostic machine" program. Once the host level 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 UDA50-A or KDA50-Q are to be tested and printing error messages.

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

HOST PROGRAM CONTROLLER AT csr RUNTIME hh:mm:ss

The host program (PDP-11) detected the error. CONTROLLER AT csr identifies the address of the UDA50-A or KDA50-Q being tested. It may be omitted if the error is not specific to one controller.

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

The "diagnostic machine" program loaded in test 1 detected the error.

Sample error message:

```
CZUDI DVC FTL ERR  00021 ON UNIT 00 TST 001 SUB 003 PC: 021062 - general message
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME 0:00:12             - basic message
CONTROLLER RESIDENT DIAGNOSTICS DETECTED FAILURE                \
  SA REGISTER CONTAINS 104041                                    )- extended
REPLACE CONTROLLER PROCESSOR MODULE                             /       message
```

Some 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 printed unless the "IER", "IBE" or "IXE" flags are set. These messages are for informational purposes only and their contents are self explanitory.

Sample informational message:

```
UNIT 0  CONTROLLER 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 lower case words. These include program counters and runtime. Other numbers, such as unit number, drive number, UDA50-A or KDA50-Q 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) -

3.2.1.1 00002 - TWO UNITS SELECT SAME DRIVE -

```
CZUDI SYS FTL ERR  00002 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
INVALID ANSWERS GIVEN 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.

3.2.1.2 00003 MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER -

```
CZUDI SYS FTL ERR  00003 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS
MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER
```

Up to four physical disk drives can be attached to a UDA50-A or KDA50-Q at one time. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

3.2.1.3 00004 NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED -

```
CZUDI SYS FTL ERR  00004 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  RUNTIME hh:mm:ss
NOT ENOUGH ROOM IN MEMORY TO TEST THE UNITS SELECTED
RESTART PROGRAM 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.

3.2.1.4 00006 - TABLE INCONSISTANCY ERROR -

```
CZUDI SYS FTL ERR  00006 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  RUNTIME hh:mm:ss
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 diag- nostic. Load and start your program again.

3.2.1.5 00014 - CONTROLLER IS NOT SUPPORTED BY DIAGNOSTIC PROGRAM -

```
CZUDI DVC FTL ERR  00014 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
CONTROLLER IS NOT SUPPORTED BY THIS DIAGNOSTIC PROGRAM.
THIS PROGRAM REQUIRES A UDA50A (MODEL 6, MICROCODE VERSION AT LEAST 3),
OR A KDA50-Q (MODEL 13, MICROCODE VERSION AT LEAST 0)
CONTROLLER REPORTED MODEL CODE xx AND MICROCODE VERSION xx
```

This diagnostic program will only test UDA50-A (modules M7485-6) or KDA50-Q (modules M7164-5) controllers. UDA50 (modules M7161-2) controllers will not be tested by this diagnostic controller and should be replaced. If the program detects that the controller being tested is not a UDA50-A or a KDA50-Q, it will not be tested. If the microcode version of the controller is not at the current revision level, the test will proceed, but accurate results are not guarenteed.

104000 - Fatal sequencer error
104040 - D PROC ALU test error
104041 - D PROC ROM parity test error / Timeout test error
105102 - D PROC no board 2 error / D PROC control reg test error / D PROC RAM parity error
105105 - D PROC RAM buffer error
105152 - D PROC SDI error
105153 - D PROC write mode, wrap SERDES 16 error
105154 - D PROC read mode, SERDES 16, 10 RSGEN and ECC error
106040 - U PROC ALU error / DFAIL test error / Unexpected trap
106041 - U PROC control reg test error
106042 - U PROC parity error set erroneously / CROM parity error
106047 - U PROC Constant ROM error with D proc running SDI test
106055 - Unexpected trap - abnormal termination of diagnostics
106071 - U PROC ROM error
106072 - U PROC ROM parity test error
106200 - STEP 1 data error (MSB wasn't set)
107103 - U PROC RAM parity error
107107 - U PROC RAM buffer error
107115 - Board #2 test count was wrong
112300 - STEP 2 INIT error
122240 - DMA test error
122300 - STEP 3 INIT error
142300 - STEP 4 INIT error

3.2.1.6 00030 - FATAL ERROR REPORTED BY CONTROLLER -

```
CZUDI DVC FTL ERR  00030 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
FATAL ERROR REPORTED BY CONTROLLER
SA REGISTER CONTAINS  100004
```

A message from the controller firmware reports an unexpected failure.

An error code is presented in the SA register. Here is a list of the codes and their meanings:

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

L2

SEQ 0024

3.2.1.7 00031 - NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES -

```
CZUDI DVC FTL ERR  00031 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
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.

3.2.1.8 00032 - UNKNOWN REQUEST RECEIVED FROM DM PROGRAM -

```
CZUDI DVC FTL ERR  00032 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
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 bus or either one of the controller 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.

3.2.1.9 00033 - RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA -

```
CZUDI DVC FTL ERR  00033 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA
EITHER CONTROLLER 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
   000000 000000             000000 000000
```

The host program inspected the response packet which was given by to controller. 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 controller 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).

3.2.1.10 00035 - DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE -

```
CZUDI DVC FTL ERR  00035 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE
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 currently running DM program sent a message to the controller referencing a drive which doesn't have an entry in the host program's drive data tables. The message sent to the host program is also included in the error message. This error indicates either a software

or hardware error. Re-load the diagnostic program and run TEST 1 to diagnose the failure.

3.2.1.11 00036 - NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS -

```
CZUDI DVC FTL ERR  00036 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS
WHILE LOADING DM PROGRAM
```

After a DM program has been sent to the controller, 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.

3.2.1.12 00038 - MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS -

```
CZUDI DVC FTL ERR  00038 ON UNIT 00 TST 001 SUB 002 PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS
CHECK BUS SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE
OR BUS
OR REPLACE CONTROLLER PROCESSOR MODULE
```

A non-existant memory error occurred when the host program tried to access the controller IP and SA registers while in subtest 2 of test 1. The controller is at another CSR address (check the bus selection switches) or the processor module is broken or the bus is broken.

3.2.1.13 00040 - FATAL CONTROLLER ERROR -

```
CZUDI DVC FTL ERR  00040 ON UNIT 00 TST 001 SUB 000 PC: hostpc
HOST PROGRAM CONTROLLER AT 172150  RUNTIME hh:mm:ss
FATAL CONTROLLER ERROR
USE DIAGNOSTIC ZUDHAO TO DIAGNOSE FAILURE
```

This error message will be printed when a controller timeout or error occurs during controller initialization. It is not the intent of this diagnostic to diagnose controller errors. If this message is printed, use the diagnostic program ZUDHAO to diagnose the controller failure.

3.2.2 DM PROGRAM INFORMATIONAL MESSAGES -

These messages inform the operator of events occurring that may be of intrest.

3.2.2.1 ECC CORRECTABLE DATA -

UNIT unit CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
ECC CORRECTABLE DATA EXISTS IN type bn
number_of_errors OUT OF A MAXIMUM OF ecc_maximum SYMBOLS WERE CORRECTED
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder

number_of_errors:
The number of symbols that were corrected using the ECC correction algorithim.
ecc_maximum:
The maximum number of ECC errors that can be corrected (an SDI DRIVE CHARACTERISTIC) without error.

The above message occure 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.

3.2.2.2 INITIAL WRITE COMPLETE -

UNIT unit CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
INITIAL WRITE COMPLETE

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

3.2.2.3 READ ONLY DRIVE, INITIAL WRITE WILL NOT BE PERFORMED -

UNIT unit CONTROLLER AT caddr 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!!

3.2.2.4 THE AREA BEING TESTED HAS N BYTES/SECTOR -

UNIT unit CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
THE AREA BEING TESTED HAS bytes_per_sector BYTES/SECTOR

bytes_per_sector:
Number of bytes contained in one sector on the drive being tested. This will be 512 for 16 bit drives and 576 for 18 bit drives.

THIS MESSAGE WILL ONLY APPEAR ON MANUFACTURING SOFTWARE. NO FIELD DIAGNOSTIC WILL DISPLAY THIS MESSAGE.

Manufacturing has special TEST 4 software that allows TEST 4 to run on both 16 and 18 bit drives. To enable the operator to know what kind of drive the test is being run on, the above message will print out at the start of the test.

3.2.2.5 DEVICE FATAL WILL CAUSE THE FOLLOWING DRIVES TO BE DROPPED -

UNIT unit CONTROLLER AT caddr 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 a 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 CONTROLLER IS ON A SYSTEM, THIS MESSAGE MAY NOT IMMEDIATELY FOLLOW THE DEVICE FATAL IF AN ERROR HAPPENS AT THE SAME TIME ON ANOTHER CONTROLLER.

3.2.3 DM PROGRAM ERROR MESSAGES (04000 TO 04999) -

3.2.3.1 4001 - ATTN ASSERTED DURING SEEK -

```
CZUDI SFT ERR  04001 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
ATTN ASSERTED DURING SEEK
SEEK FROM GRP group CYL cylinder TO GRP group CYL cylinder
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.2 4002 - ATTN ASSERTED UNEXPECTEDLY -

```
CZUDI SFT ERR  04002 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
ATTN ASSERTED UNEXPECTEDLY, ASYNC DRIVE ERROR OR LOGGABLE
INFORMATION
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

This is an asynchronous drive 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. The operator must look at the status returned to determine the error that occurred.

See retry/recovery section for recovery details.

3.2.3.3 4003 - SEEK DID NOT COMPLETE -

```
CZUDI SFT ERR  04003 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.4 4004 - RCT AREA CORRUPTED -

```
CZUDI HRD ERR  04004 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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

CZUDI HRD ERR  04004 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.5 4005 - HEADER NOT FOUND DURING WRITE -

```
CZUDI HRD ERR  04005 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.6 4006 - SELECT TRACK AND WRITE NOT SENT -

```
CZUDI SFT ERR  04006 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
SELECT TRACK AND WRITE LEVEL 1 CMD NOT SENT
ATTEMPT attempt
type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
ORIGIN OF SEEK: GRP group CYL cylinder
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

Select track and read or write not executed occurs when the controller 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 controller 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.

3.2.3.7 4007 - ECC DETECTED ERROR -

```
CZUDI SFT ERR  04007 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.8 4008 - ECC DETECTED ERROR, BUT CORRECTION FAILED -

```
CZUDI SFT ERR  04008 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.9 4009 - ECC CORRECTIONS EXCEED THERSHOLD -

```
CZUDI SFT ERR  04009 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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. The correction threshold is a SDI DRIVE CHARACTERISTIC; at startup, the controller asks the drive how many bits are ALLOWED to be corrected. If, after correction, the number of corrections exceed this threshold, error 40009 results.

See retry/recovery section for recovery details.

3.2.3.10 4010 - ECC CORRECTION SUCCEEDED, BUT EDC DETECTS ERROR -

```
CZUDI SFT ERR  04010 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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
EDC COMPUTED edc_computed
EDC READ     edc_read
```

edc_computed:
This is the EDC in octal computed over the sector as it resided in the controller's memory.

edc_read:
This is the EDC in octal that was found written at the end of the sector.

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) a buffer RAM problem within the controller.

See retry/recovery section for recovery details.

3.2.3.11 4011 - READ DID NOT SUCCEED ON ANY RECOVERY LEVEL -

```
CZUDI HRD ERR  04011 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
READ DID NOT SUCCEED ON ANY RECOVERY LEVEL
type bn
GRP group CYL cylinder
```

Error 4011 occurs when retries are enabled, and TEST 4 has tried to read a sector using all retries on all levels of error recovery. See ECC and EDC retries in the retry/recovery section.

3.2.3.12 4012 - DATA COMPARISON FAILED -

```
CZUDI HRD ERR  04012 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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

CZUDI HRD ERR  04012 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 occurr in the fist word of the sector. The following is the first word of the sector (octal) 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.

3.2.3.13 4013 - DRIVE NOT ONLINE TO CONTROLLER, AND NOT SPINABLE -

CZUDI DVC FTL ERR 04013 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
DRIVE NOT ONLINE TO CONTROLLER, AND NOT SPINABLE

If a drive droppe offline while ing 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.

3.2.3.14 4014 - UNABLE TO COMPLETE SEEK -- TRIED 3 TIMES -

```
CZUDI DVC FTL ERR  04014 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.15 4015 - SEEK REQUIRED N RETRIES BEFORE COMPLETING -

```
CZUDI SFT ERR  04015 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.16 4016 - ERRORS DURING DRIVE INITIALIZATION AND SETUP -

```
CZUDI DVC FTL ERR  04016 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
ERRORS DURING DRIVE INITIALIZATION AND SETUP
THIS CONTROLLER AND ALL DRIVES ATTACHED WILL BE REMOVED FROM TESTING
```

If any errors occur during drive and test initialization, DRIVES ATTACHED TO THE CONTROLLER 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 CONTROLLER INITIALIZATION. This error is unaffected by the operator inhibiting the dropping of units.

3.2.3.17 4017 - NO VALID STATE FROM DRIVE -

```
CZUDI DVC FTL ERR  04017 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
NO VALID STATE FROM DRIVE
NO DRIVE CLOCKS
```

```
CZUDI DVC FTL ERR  04017 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 drive transmitter or controller 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.

3.2.3.18 4018 - ATTEMPT TO WRITE ON WRITE PROTECTED DRIVE -

```
CZUDI DVC FTL ERR  04018 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
ATTEMPT TO WRITE ON WRITE PROTECTED DRIVE
ERROR CODE RETURNED FROM CONTROLLER: code
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

code:

The error (in octal) returned to TEST 4 from the controller when TEST 4 attempted to write on the write protected drive.

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

code	error
2	SELECT TRACK AND WRITE LEVEL 1 CMD NOT SENT
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 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.

3.2.3.19 4019 - HEADER NOT FOUND DURING READ -

```
CZUDI HRD ERR  04019 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 can occur only when TEST 4 is reading a DBN, RBN or RCT LBN.

DBN space is for diagnostic use only, and bad blocks in this area are merely written with a bad header code; TEST 4 will call out the 'missing' block with this error, but this condition IS NOT an error.

The RCT LBN space is protected from bad blocks with a multi-copy structure. No blocks in the RCT LBN space are revectored. TEST 4 will call out the 'missing' block with this error, but this condition IS NOT and error. It is an error if all copies of the RCT cannot be read, in which case a group of these errors will be followed by a 4040 error. See error 4040 for more information.

If this error occurrs on a RBN, this is a definate hard error. This is because bad LBN's are revectored to RBN's, so the RBN that the LBN is revectored to 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.

3.2.3.20 4020 - SELECT TRACK AND READ LEVEL 1 CMD NOT SENT -

```
CZUDI SFT ERR  04020 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
SELECT TRACK AND READ LEVEL 1 CMD NOT SENT
ATTEMPT attempt
type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
ORIGIN OF SEEK: GRP group CYL cylinder
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

Select track and read or write not executed occurs when the controller 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 controller 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.

3.2.3.21 4021 - DRIVE NOT FORMATTED IN 512 BYTE MODE -

```
CZUDI DVC FTL ERR  04021 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DRIVE NOT FORMATTED IN 512 BYTE MODE -- UNABLE TO TEST
FCT BLOCK ZERO MODE WORD: mode

*** THIS PACK HAS AN INVALID FORMAT AND CANNOT BE USED ***
```

mode:
The mode word found on the drive's FCT block zero

Error 4021 occurs when TEST 4 Finds that the mode word found in FCT block zero 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. A disk with this error must be re-formatted.

3.2.3.22 4021 - DRIVE NOT FORMATTED IN 512 OR 576 BYTE MODE -

```
CZUDI DVC FTL ERR  04021 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DRIVE NOT FORMATTED IN 512 OR 576 BYTE MODE -- UNABLE TO TEST
FCT BLOCK ZERO MODE WORD: mode

*** THIS PACK HAS AN INVALID FORMAT AND CANNOT BE USED ***
```

mode:
The mode word found on the drive's FCT block zero

THIS ERROR WILL ONLY APPEAR ON MANUFACTURING SOFTWARE. NO FIELD DIAGNOSTIC WILL DISPLAY THIS ERROR.

Error 4021 occurs when TEST 4 Finds that the mode word found in FCT block zero is not the 512 or 576 byte mode word (126736 octal for 512. 074161 for 576 bytes/sector). See DEC STANDARD 166 "FCT Structure". Inhibiting the dropping of units has no effect on this error. A disk with this error must be re-formatted.

3.2.3.23 4022 - COULD NOT READ FCT BLOCK ZERO -

```
CZUDI DVC FTL ERR  04022 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
COULD NOT READ FCT BLOCK ZERO

*** THIS PACK HAS AN INVALID FORMAT AND CANNOT BE USED ***
```

Error 4022 occurs when TEST 4 is unable to read ANY copy of FCT block zero. See DEC STANDARD 166 "FCT Structure". Inhibiting the dropping of units has no effect on this error. A disk with this error must be re-formatted.

3.2.3.24 4023 - UNABLE TO CONTINUE TESTING -

```
CZUDI DVC FTL ERR  04023 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
UNABLE TO CONTINUE TESTING
PORT SWITCH OUT
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

If, during testing, the operator disables the drive port that TEST 4 is using by popping out the drive's 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).

```
CZUDI DVC FTL ERR  04023 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
UNABLE TO CONTINUE TESTING
RUN/STOP SWITCH OUT
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

If, during testing, the operator pops out the drive's 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).

```
CZUDI DVC FTL ERR  04023 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
UNABLE TO CONTINUE TESTING
SPINDLE DROPPED READY
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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).

3.2.3.25 4024 - EDC DETECTED ERROR BUT ECC DID NOT -

```
CZUDI SFT ERR  04024 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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_computed
EDC READ     edc_read

edc_computed:
     This is the EDC in octal computed over the sector as  it  resided
     in the controller's memory.

edc_read:
     This is the EDC in octal that was found written at the end of the
     sector.

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) a
buffer RAM problem within the controller, or 3) The error is such that
the ECC really doesn't detect an error...  Case 3 is very unlikely.

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

3.2.3.26 4025 - WRITE ATTEMPTED MAXIMUM TIMES -

```
CZUDI HRD ERR  04025 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.
```

3.2.3.27 4026 - READ ATTEMPTED MAXIMUM TIMES -

```
CZUDI HRD ERR  04026 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.28 4028 - BOTH READ ONLY <AND> WRITE ONLY BITS SET -

```
CZUDI DVC FTL ERR  04028 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.29 4029 - HEADER NOT FOUND DURING ERROR LEVEL RECOVERY -

```
CZUDI SFT ERR  04029 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
HEADER NOT FOUND DURING ERROR LEVEL RECOVERY
ATTEMPT attempt
type bn
SECTORS FROM INDEX sector TRK track GRP group CYL cylinder
ORIGIN OF SEEK: GRP group CYL cylinder
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

This error occurs during error recovery when the recovery methods cause the header to be unable to be read.

It is quite possible that the recovery methods (such as moving the heads off-track) will cause the header to be un-readable. Note that this does NOT destroy the header, the drive has modified its read-write path in such a way that the header can't be read at this instant. In this case, the correct action IS NOT to see if the sector is revectored. All retries and retry levels should be attempted first.

This error occurs ONLY if error level recovery is allready in progress when the 'HEADER NOT FOUND' error appears.

3.2.3.30 4030 - ERROR DETECTED AFTER DRIVE WAS SPUN DOWN -

```
CZUDI HRD ERR  04030 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
ERROR DETECTED AFTER DRIVE WAS SPUN DOWN
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

THIS ERROR WILL ONLY APPEAR ON MANUFACTURING SOFTWARE. NO FIELD DIAGNOSTIC WILL DISPLAY THIS ERROR.

Some manufacturing software causes the drive being tested to be spun down, then up before exercising begins. If, after spinning the drive down, a drive error is detected the above message will be displayed.

3.2.3.31 4034 - SERDES OVERRUN ERROR DURING READ -

```
CZUDI SFT ERR  04034 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.32 4035 - DATA OR STATE CLOCK TIMEOUT DURING READ -

```
CZUDI SFT ERR  04035 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

The loss of drive clock occurs when the controller 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.

3.2.3.33 4036 - DATA SYNC TIMEOUT DURING READ -

```
CZUDI SFT ERR  04036 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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

See retry/recovery section for recovery details.

3.2.3.34 4037 - R/W RDY DROPPED BEFORE/DURING READ -

```
CZUDI SFT ERR  04037 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
R/W RDY DROPPED BEFORE/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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.35 4038 - RCVR RDY DROPPED BEFORE/DURING READ -

```
CZUDI SFT ERR  04038 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
RCVR RDY DROPPED BEFORE/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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

The loss of drive receiver ready is detected when the controller is

trying to send out a real-time read or write command.

See retry/recovery section for recovery details.

3.2.3.36 4040 - ALL COPIES OF RCT READ WITH ERROR -

```
CZUDI HRD ERR  04040 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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

CZUDI HRD ERR  04040 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.37 4041 - COULD NOT FIND REPLACEMENT -

```
CZUDI HRD ERR  04041 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
COULD NOT FIND REPLACEMENT FOR
LBN THAT WAS REVECTORED
LBN TO REPLACE bn

CZUDI HRD ERR  04041 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
COULD NOT FIND REPLACEMENT FOR
LBN WITH HEADER NOT FOUND
LBN TO REPLACE bn
```

Error 4041 only occure 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).

3.2.3.38 4042 - TIMEOUT WAITING FOR SECTOR OR INDEX PULSE -

```
CZUDI DVC FTL ERR  04042 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
TIMEOUT WAITING FOR SECTOR OR INDEX PULSE
GRP group CYL cylinder
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

Error 4042 occurs when the controller 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.

3.2.3.39 4044 - SEEK OR HEAD SELECT ERROR DETECTED DURING WRITE -

```
CZUDI SFT ERR  04044 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

See error 4045 for description.

See retry/recovery section for recovery details.

3.2.3.40 4045 - SEEK OR HEAD SELECT ERROR DETECTED DURING READ -

```
CZUDI SFT ERR  04045 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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 miseeek 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.

3.2.3.41 4047 - DATA OR STATE CLOCK TIMEOUT DURING WRITE -

```
CZUDI SFT ERR  04047 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

The loss of drive clock occure when the controller 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.

3.2.3.42 4048 - R/W RDY DROPPED BEFORE/DURING WRITE -

```
CZUDI SFT ERR  04048 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
R/W RDY DROPPED BEFORE/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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.43 4049 - RCVR RDY DROPPED BEFORE/DURING WRITE -

```
CZUDI SFT ERR  04049 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
RCVR RDY DROPPED BEFORE/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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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

See retry/recovery section for recovery details.

3.2.3.44 4050 - BEGIN/END SET STARTING BLOCK NUMBER GREATER THAN ENDING BLOCK NUMBER -

```
CZUDI DVC FTL ERR  04050 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.45 4051 - THE BEGIN/END SETS OVERLAP -

```
CZUDI DVC FTL ERR  04051 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.46 4052 - BEGIN/END SET ENDING BLOCK NUMBER EXCEEDS MAXIMUM -

```
CZUDI DVC FTL ERR  04052 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.47 4053 - DUPLICATE BAD BLOCKS -

```
CZUDI DVC FTL ERR  04053 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.48 4054 - BAD BLOCK NUMBER EXCEEDS MAXIMUM -

```
CZUDI DVC FTL ERR  04054 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.49 4055 - STARTING CYLINDER GREATER THAN ENDING CYLINDER -

CZUDI DVC FTL ERR 04055 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug 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.

3.2.3.50 4056 - RANDOM AND SEQUENTIAL SEEKS CANNOT BE MIXED WITHIN A UNIT -

CZUDI DVC FTL ERR 04056 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug 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.

3.2.3.51 4057 - OVERFLOW WHEN CALCULATING THE L/DBN FROM THE GIVEN CYLINDER -

CZUDI DVC FTL ERR 04057 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug 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.

3.2.3.52 4058 - TRACK EXCEEDS MAXIMUM FOR DEVICE -

3.2.3.53 4058 - GROUP EXCEEDS MAXIMUM FOR DEVICE -

```
CZUDI DVC FTL ERR  04058 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

```
CZUDI DVC FTL ERR  04058 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.54 4059 - TWO IDENTICAL TRACKS -

3.2.3.55 4059 - TWO IDENTICAL GROUPS -

```
CZUDI DVC FTL ERR  04059 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

```
CZUDI DVC FTL ERR  04059 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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.

3.2.3.56 4062 - DBN COMPUTED FROM END CYLINDER GIVEN EXCEEDS MAXIMUM DBN NUMBER -

```
CZUDI DVC FTL ERR  04062 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 due to an operator 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.

```
CZUDI DVC FTL ERR  04062 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 due to an operator 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.

3.2.3.57 4063 - REAL TIME STATE RECEIVE ERROR DURING WRITE -

```
CZUDI SFT ERR  04063 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.58 4064 - REAL TIME STATE RECEIVE ERROR DURING READ -

```
CZUDI SFT ERR  04064 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  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 state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.59 4068 - UNKNOWN ERROR CODE DURING WRITE -

```
CZUDI HRD ERR  04068 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
UNKNOWN ERROR CODE DURING WRITE
ERROR CODE RETURNED error_code
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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

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

See retry/recovery section for recovery details.

3.2.3.60 4069 - UNKNOWN ERROR CODE DURING READ -

```
CZUDI HRD ERR  04069 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
UNKNOWN ERROR CODE DURING READ
ERROR CODE RETURNED error_code
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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

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

See retry/recovery section for recovery details.

3.2.3.61 4070 - TIMEOUT OF SEND -

```
CZUDI SFT ERR  04070 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
TIMEOUT OF SEND
command_type
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.62 4071 - TIMEOUT OF RECEIVE -

```
CZUDI SFT ERR  04071 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
TIMEOUT OF RECEIVE
command_type
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0
```

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.

3.2.3.63 4072 - FIRST WORD RECEIVED WAS NOT START FRAME -

```
CZUDI SFT ERR  04072 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
FIRST WORD RECEIVED WAS NOT START FRAME
command_type
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0

command_type:
     See section following error 4078 for a description
```

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

See retry/recovery section for recovery details.

3.2.3.64 4073 - FRAMING ERROR ON LEVEL 0 RECEIVE -

```
CZUDI SFT ERR  04073 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
FRAMING ERROR ON LEVEL 0 RECEIVE
command_type
REAL TIME STATE state
STATUS (R TO L): word6  word5  word4  word3  word2  word1  word0

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 controller asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or receiver on the controller.

See retry/recovery section for recovery details.

3.2.3.65 4074 - CHECKSUM ERROR ON LEVEL 0 RECEIVE -

CZUDI SFT ERR 04074 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
CHECKSUM ERROR ON LEVEL 0 RECEIVE
command_type
REAL TIME STATE state
STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0

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 controller receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable.

See retry/recovery section for recovery details.

3.2.3.66 4075 - BUFFER SIZE SMALLER THAN LEVEL 2 RESPONSE -

CZUDI SFT ERR 04075 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
BUFFER SIZE SMALLER THAN LEVEL 2 RESPONSE
command_type
REAL TIME STATE state
STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0

command_type:
See section following error 4078 for a description

The buffer size set aside for the reponse 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.

3.2.3.67 4076 - RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED -

CZUDI SFT ERR 04076 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug 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 state
STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0

command_type:
See section following error 4078 for a description

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

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

This is caused by receiving an UNSUCCESSFUL reponse 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 reponse error (or reason).

See retry/recovery section for recovery details.

3.2.3.68 4077 - DRIVE NEVER DEASSERTED RECEIVER READY AFTER LEVEL 2 SEND -

CZUDI SFT ERR 04077 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
DRIVE NEVER DEASSERTED RECEIVER READY AFTER LEVEL 2 SEND
command_type
REAL TIME STATE state
STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0

command_type:
See section following error 4078 for a description

After a successful send of a level 2 command and before receiving the response, TEST 4 checks the drive's receiver ready to make sure it has been deasserted. If it hasn't, error 4077 is printed.

See retry/recovery section for recovery details.

3.2.3.69 4078 - UNKNOWN ERROR CODE RETURNED FROM LEVEL 2 RECEIVE -

CZUDI SFT ERR 04078 ON UNIT unit TST 001 SUB 000 PC: hostpc
DISK EXERCISER DM PC:dmpc CONTROLLER AT caddr DRIVE plug RUNTIME hh:mm:ss
UNKNOWN ERROR CODE RETURNED FROM LEVEL 2 RECEIVE
command_type
ERROR CODE RETURNED error_code
REAL TIME STATE state
STATUS (R TO L): word6 word5 word4 word3 word2 word1 word0

command_type:

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

The unknown error code occurs when the controller returns an error code from an operation that TEST 4 does not recognize. Possible controller 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 command_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 controller that the initialization erorr occured on. See error number 4016.

3.2.4 SPECIAL DEVICE FATAL (05000) -

```
CZUDI DVC FTL ERR 05000 ON UNIT 00 TST 001 SUB 000 PC: hostpc
DISK zzzzzzzz DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
UNABLE TO FIND REQUESTED DRIVE FOR TESTING
THE FOLLOWING IS VISIBLE ON THE PORTS
PORT 0 -- description
PORT 1 -- description
PORT 2 -- description
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 or test 3 on any of the ports. A description of what was on 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 controller 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 controller from the drive was not a valid message start frame.

FRAMING ERROR ON LEVEL 0 RECEIVE
The device and the controller 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 controller asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or controller 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 controller 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 [further explanation]
A drive was found at the end of the cable. A further explanation may be presented. These further explanations are:

DRIVE NOT AVAILABLE TO THIS CONTROLLER
The drive was found but is not available to this controller. 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.

4.0 DM PROGRAM RETRY AND RECOVERY METHODS

4.1 ECC ERRORS

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

Retry/Recovery - The controller 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 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.

4.2 EDC ERRORS

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

This error is indicative of a controller 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 controller 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 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.

4.3 SDI LEVEL 2 AND ASYNCHRONOUS ERRORS

The SDI level 2 errors are as follows:

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

4.3.1 PACKET ACKNOWLEDGE FAILURE -

TIMEOUT OF SEND
TIMEOUT OF RECEIVE

The timeout of send occurs when the controller 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 - Controller mode - 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, 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.

4.3.2 LEVEL 2 COMMAND ERROR RESPONSES -

RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED

A level 2 error response is when a command has successfully been sent to the drive and an error-free level 2 response has been received which has a UNSUCCESSFUL or unrecognized opcode.

4.3.2.1 "DE" BIT SET -

RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED

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

Retry/Recovery - Controller mode - 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 error 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, 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.

4.3.2.2 "PE" OR "RE" BIT SET -

RESPONSE OF LEVEL 2 CMD NOT AS EXPECTED

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

Retry/Recovery - Controller mode - 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 controller 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, 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.

4.3.3 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
DRIVE NEVER DEASSERTED RECEIVER READY AFTER LEVEL 2 SEND
UNKNOWN ERROR CODE RETURNED FROM LEVEL 2 RECEIVE (hard error)

The first word not start frame error is caused when the controller does not see a valid message start frame as the first frame received from the drive. The framing error is caused by the controller 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 reponse was not large

enough for the response received. The unknown error code is returned when the controller returns an error code that TEST 4 does not recognize. These errors are grouped together because their recoveries are the same.

Retry/Recovery - Controller mode - 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, 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.

4.3.4 SEEK COMPLETE TIMEOUT -

ATTN ASSERTED DURING SEEK
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 - Controller mode - 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, 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.

4.3.5 ASYNCHRONOUS DRIVE ERRORS -

ATTN ASSERTED UNEXPECTEDLY, ASYN DRIVE ERROR OR LOGGABLE INFORMATION

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 - Controller mode - 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 controller mode and TEST 4 for this type of error.

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

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

4.3.5.1 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:

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

4.3.6 HEADER NOT FOUND (HEADER COMPARE ERROR) -

HEADER NOT FOUND DURING (read/write)

This error occure 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 - Controller mode and TEST 4 - Failure to find the desired header in two rotations of the disk will cause 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, TEST 4 will perform the desired operation on the revectored block. Enabling/disabling retries has no effect on this operation.

Recovery success - No error is reported or counted.

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

4.3.7 SEEK OR HEAD SELECT ERROR (POSITIONER ERROR) -

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

This error occure 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 - Controller mode - 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 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.

4.3.8 DATA SYNC TIMEOUT ERROR -

DATA SYNC TIMEOUT DURING READ

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

Retry/Recovery - Controller mode - 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 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.

4.3.9 DATA OR STATE CLOCK TIMEOUT (LOSS OF DRIVE CLOCK) OR 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 occure when the controller 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 controller is trying to send out a real-time read or write command. Unable to select track and read or write occurs when the controller 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 controller 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 - Controller mode - 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 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.

4.3.10 READ/WRITE READY DROPPED (LOSS OF DRIVE READ/WRITE READY) OR SERDES - OVERRUN ERROR OR 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 controller returns an error code that TEST 4 does not recognize. They are grouped together because their recoveries are the same.

Retry/Recovery - Controller mode - 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 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.

5.0 DEC STANDARD 166 EXCERPTS

DEC standard 166 is also known as the Digital standard disk format (DSDF). It specifies the exact format to be found on any RA family disk.

5.1 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.

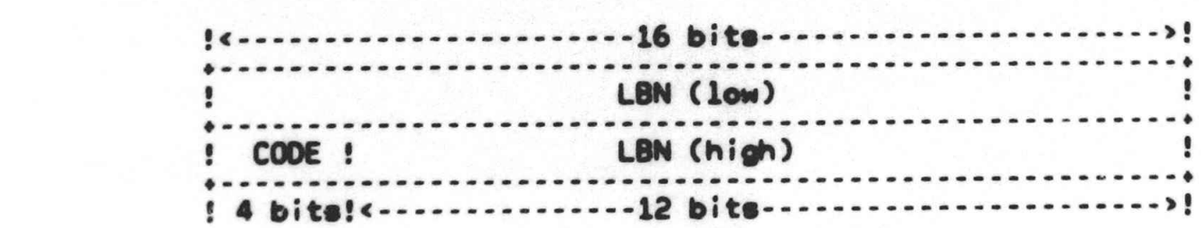
5.2 Replacement And Caching Table Format

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.

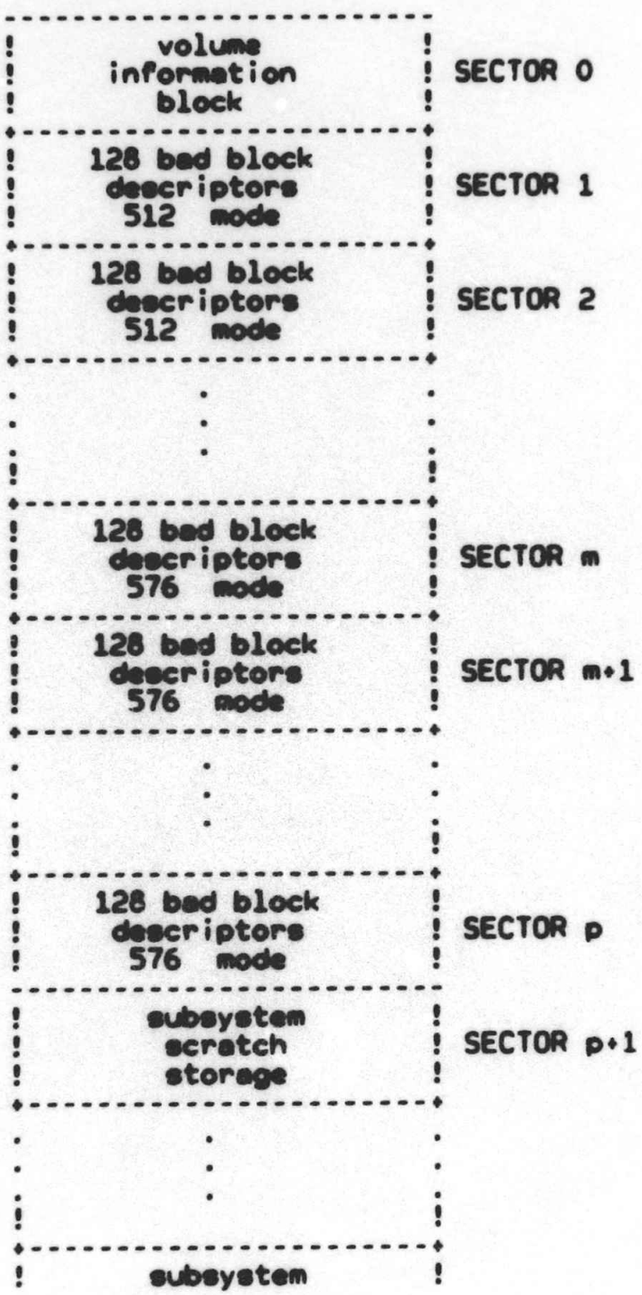
Embedded-controllers with no distinction between primary and secondary RBN's must use:

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.

5.3 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:



```
                 !        scratch        ! SECTOR Fct-1
                 !        storage        !
                 +-----------------------+
```

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
```

5.4 FCT SECTOR 0 CONTENTS

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
number of used entries in 512 table (high)	WORD 15

number of used entries in 576 table (low)	WORD 16
number of used entries in 576 table (high)	WORD 17
XBN of scratch area in this copy (low)	WORD 18
XBN of scratch area in this copy (high)	WORD 19
size of scratch area in this copy	WORD 20
zeros	
zeros	WORD 255

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.

6.0 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). 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. The report will contain statistics on each drive for the current pass of the test; for example:

TEST 1 IN PROGRESS RUN TIME 2:24:10

UNIT	DRIVE	SERIAL-NUMBER	SEEKS X1000	MBYTES READ	MBYTES WRITTEN	HARD ERRORS	SOFT ERRORS	ECC
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 the test. Multiply value by 1000.

MBYTES READ — The number of mega-bytes (million bytes) read by this drive during this pass of the test. 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 the test.

HARD ERRORS — The number of hard error reports printed for this drive during this pass of the test. 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 the test. 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).

7.0 TEST SUMMARIES

The controller Host Resident Diagnostic consists of one PDP-11 diagnostic supervisor program that runs in the PDP-11 processor and one program that runs in the controller's buffer memory through an interpreter called the "diagnostic machine" which resides in the controller. The PDP-11 program mainly is responsible for downline loading the "diagnostic machine" program into the controller and starting its 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.

7.1 TEST 1 - DISK DRIVE EXERCISER

The purpose of test 1 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 1:

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 CONTROLLER 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 con- verted 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.

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.

ENTER TEST AREA DESIRED:
0 - ENTIRE DISK AREA
1 - SPECIC BEGIN/END SETS
2 - SPECIC TRACKS & CYLINDERS
3 - SPECIC GROUPS & CYLINDERS
4 - SPECIC CYLINDERS 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.

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.

The test will then initialize each controller selected for testing and downline load a "diagnostic machine" program into each controller. 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 controllers have been started, the PDP-11 program responds to requests from all controllers.

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 controllers 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 the test.

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, the test will end (unless the error log is enabled).

When the end of the test occurs, the accumulated statistics for each drive is printed. This statistical report can be printed at any time during testing by typing control-C then the PRINT command.

The data patterns used by the test 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) 8888

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 the test 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

CHANGE HW (L)  ? N

CHANGE SW (L)  ? Y

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

```
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 CONTROLLER AT 172150 DRIVE 0

NUMBER OF BAD BLOCKS (D) 0 ? 2

BAD BLOCK (A) ? 234

BAD BLOCK (A) ? 8900

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

ENTER TEST AREA DESIRED:
   0 - ENTIRE DISK AREA
   1 - SPECIC BEGIN/END SETS
   2 - SPECIC TRACKS & CYLINDERS
   3 - SPECIC GROUPS & CYLINDERS
   4 - SPECIC CYLINDERS
 (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 ?
8
```

```
                        .NLIST BEX,CND
                        ;++
                        ;       REVISION HISTORY:
                        ;       REV. A - JFM - 12-SEP-1984
                        ;               THIS PROGRAM HAS BEEN ADAPTED FROM CZUDCO, REVISION E.
                        ;
                        ;--
                        ;++
                        ;               M A C R O   D E F I N I T I O N   S E C T I O N
                        ;--

                        ;++
                        ;       PUSH - PUT DATA ON THE STACK
                        ;
                        ;       ARGUMENTS:
                        ;               A - DATA TO BE PUT ON THE STACK
                        ;--

                        .MACRO  PUSH    A
                        .IRP    B,<A>
                                MOV     B,-(SP)         ; PUSH B ON STACK
                        .ENDM
                        .ENDM   PUSH

                        ;++
                        ;       POP - REMOVE DATA FROM THE STACK
                        ;
                        ;       ARGUMENTS:
                        ;               A - LOCATION TO PUT THE DATA REMOVED FROM THE STACK
                        ;--

                        .MACRO  POP     A
                        .IRP    B,<A>
                                MOV     (SP)+,B         ; POP STACK INTO B
                        .ENDM
                        .ENDM   POP

                        ;++
                        ;       ASSUME - CHECK VALIDITY OF PROGRAM ASSUMPTIONS
                        ;--
                        .MACRO  ASSUME  FIRST,CONDITION,SECOND
                        .IF CONDITION <FIRST>-<SECOND>
                                .IFF
                                .ERROR  ;BAD ASSUME OF <FIRST> CONDITION <SECOND>
                        .ENDC
                        .ENDM   ASSUME

                        ;++
                        ;       MACRO DEFINITIONS FOR GLOBAL EQUATES
                        ;
                        ;       THESE MACROS ARE USED TO DEFINE INDEXES INTO A TABLE
                        ;
                        ;       CALLING SEQUENCE MUST BE
                        ;
                        ;               TABLE
                        ;               ITEM    NAME    BYTES   COMMENT
                        ;               ITEM    NAME    BYTES   COMMENT
                        ;               ITEM    NAME    BYTES   COMMENT
```

```
                        ;               END     SIZE
                        ;
                        ;       TABLE - DESIGNATES THAT A TABLE IS ABOUT TO BE DEFINED.
                        ;       END - TERMINATES THE DEFINITION.
                        ;       ITEM - ENTRY IN THE TABLE. ANY NUMBER OF ITEM LINES CAN APPEAR.
                        ;       NAME - THE NAME OF THE SYMBOL BEING EQUATED TO THE INDEX. THE INDEX
                        ;              ALWAYS STARTS AT ZERO.
                        ;       BYTES - THE SIZE OF THE VALUE TO BE STORED AT THAT INDEX IN BYTES.
                        ;       SIZE - (OPTIONAL) THE SIZE OF THE TABLE IN BYTES.
                        ;       TINDEX - KEEPS TRACK OF THE INDEX VALUE AND WILL BE EQUAL TO THE SIZE
                        ;              OF THE TABLE AFTER THE END STATEMENT.
                        ;--

                        .MACRO  TABLE
                                TINDEX  = 0
                        .ENDM   TABLE

                        .MACRO ITEM NAME BYTES COMMENT
                                NAME=TINDEX                                     ;COMMENT
                                TINDEX=TINDEX+BYTES
                        .ENDM   ITEM

                        .MACRO END SIZE COMMENT
                                .IF NB SIZE
                                SIZE=TINDEX                                     ;COMMENT
                                .ENDC
                        .ENDM   END

                        ;++
                        ;       PRINT - 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.. PNTF, PNTB, PNTX, PNTS
                        ;--

                        .MACRO  PRINT   ARG1
                        .IF DIF <ARG1>,R0
                                MOVB    ARG1,R0                 ; STORE ARG1 IN R0 AND
                        .ENDC
                                CALL    PRINTC          ; PRINT THE CHARACTER.
                        .ENDM   PRINT

                        ;++
                        ;       PNT... - PROCESSING MACRO FOR NEXT SET OF FORMATTED MESSAGE MACROS
                        ;--

                        .MACRO  PNT...  RTN,ADR,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                        ARG.CT  = 0
                        .IRP AA,<ARG8,ARG7,ARG6,ARG5,ARG4,ARG3,ARG2,ARG1>
                                .IF NB,<AA>
                                MOV     AA,-(SP)                ; PUSH AA ON STACK
                        ARG.CT  = ARG.CT+2                      ; INCREMENT ARGUMENT COUNT
                        .ENDC
                        .ENDM
```

```
                                        JSR     R1,RTN                  ; CALL RTN PRINT ROUTINE
                                        .WORD   ADR                     ; ADDRESS OF ASCIZ STRING
                                        .WORD   ARG.CT                  ; ARGUMENT COUNT * 2
                                .ENDM   PNT...

                                ;++
                                ;       PNTF, PNTB, PNTX, PNTS - 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   PNTF

                                .MACRO  PNTB    ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                        PNT...  LPNTB ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                .ENDM   PNTB

                                .MACRO  PNTX    ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                        PNT...  LPNTX ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                .ENDM   PNTX

                                .MACRO  PNTS    ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                        PNT...  LPNTS ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                .ENDM   PNTS

                                .MACRO  PNT     ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                        PNT...  LPNT ADR ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
                                .ENDM   PNT

                                .SBTTL  PROGRAM HEADER

                                        .MCALL  SVC
000000                                  SVC                             ; INITIALIZE SUPERVISOR MACROS

                                ;++
                                ;       IF STRUCTURED MACROS ARE TO BE USED, ".MCALL STRUCT" AND "STRUCT"
                                ;       MUST BE ADDED TO INITIALIZE THE STRUCTURED MACROS.

        000000                  SVCINS= 0                               ; LIST INSTRUCTIONS, SHIFTED RIGHT
        000000                  SVCTST= 0                               ; LIST TEST TAGS, SHIFTED RIGHT
        000000                  SVCSUB= 0                               ; LIST SUBTEST TAGS, SHIFTED RIGHT
        000000                  SVCGBL= 0                               ; LIST GLOBAL TAGS, SHIFTED RIGHT
        000000                  SVCTAG= 0                               ; LIST OTHER TAGS, SHIFTED RIGHT

                                ;       THE VALUES OF THE SVC... SYMBOLS ARE ZERO TO ALIGN THE MACRO CALLS
                                ;       AND THEIR EXPANSIONS.  SETTING THE SYMBOLS TO BE MINUS-ONE WILL CAUSE
                                ;       THE EXPANSIONS TO NOT BE LISTED.  THE SYMBOLS MAY BE CHANGED AT ANY
                                ;       POINT IN THE PROGRAM.
                                ;--

000000                                  .ASECT
```

PROGRAM HEADER

```
                                        .ENABL  AMA
        002000                          .       =       2000

                                ;++
                                ;       THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                                ;       THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                                ;
                                ;       THE FOLLOWING ARE THE OPTIONAL 'HEADER' ARGUMENTS:
                                ;
                                ;       ARGUMENT        OPTION
                                ;       --------        ------
                                ;       BGNRPT          REPORT CODE
                                ;       BGNSW           SOFTWARE TABLE
                                ;       BGNSFT          SOFTWARE TABLE QUESTIONS
                                ;       BGNAU           ADD     CODE
                                ;       BGNDU           DROP CODE
                                ;       ERRTBL          ERROR TABLE
                                ;       BGNSETUP        ASSEMBLED P-TABLES
                                ;--

002000                                  POINTER BGNRPT,BGNSW,BGNSFT,ERRTBL,BGNSETUP

                                ;++
                                ;       THE "HEADER" ARGUMENTS ARE: NAME, REV, PATCH, LONGEST TEST
                                ;       TIME, TYPE, AND PRIORITY. "TYPE" = 0 FOR SEQUENTIAL DIAGNOSTIC AND = 1
                                ;       FOR EXERCISER.  "PRIORITY" SPECIFIES THE PROCESSOR PRIORITY TO BE SET
                                ;       WHEN STARTING THE DIAGNOSTIC (DEFAULT IS 0).
                                ;--

                                        HEADER  CZUDI,A,0,0,1,PRIO7     ; TEST 4
002000                          L$NAME::                        ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  125                             .ASCII /U/
002003  104                             .ASCII /D/
002004  111                             .ASCII /I/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                         ;REVISION LEVEL
002010  101                             .ASCII  /A/
002011                          L$DEPO::                        ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                        ;NUMBER OF UNITS
002012  000001                          .WORD   T$PTHV
002014                          L$TIML::                        ;LONGEST TEST TIME
002014  000000                          .WORD   0
002016                          L.$HPCP::                       ;POINTER TO H.W. QUES.
002016  025714                          .WORD   L$HARD
002020                          L$SPCP::                        ;POINTER TO S.W. QUES.
002020  026050                          .WORD   L$SOFT
002022                          L$HPTP::                        ;PTR. TO DEF. H.W. PTABLE
002022  002130                          .WORD   L$HW
002024                          L$SPTP::                        ;PTR. TO S.W. PTABLE
002024  002140                          .WORD   L$SW
002026                          L$LADP::                        ;DIAG. END ADDRESS
002026  000044'                         .WORD   L$LAST
```

```
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000001                          .WORD   1
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                          .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000340                          .WORD   PRIO7
002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                          .WORD   0
002046                          L$EXP1::                ;EXPANSION WORD
002046  000000                          .WORD   0
002050                          L$MREV::                ;SVC REV AND EDIT #
002050     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  002374                          .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  016430                          .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  000000                          .WORD   0
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  000000                          .WORD   0
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                          .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  002420                          .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                          EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  002146                          .WORD   L$ERRTBL
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  017772                          .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  021304                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  021302                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  017764                          .WORD   L$PROT
002114                          L$TEST::                ;TEST NUMBER
002114  000000                          .WORD   0
002116                          L$DLY::                 ;DELAY COUNT
002116  000000                          .WORD   0
002120                          L$HIME::                ;PTR. TO HIGH MEM
```

PROGRAM HEADER

```
        002120  000000                          .WORD   0
    203
```

```
                                .SBTTL  DISPATCH TABLE

                                ;**
                                ;       THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
                                ;       IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST. THE
                                ;       ARGUMENT OF 'DISPATCH' INDICATES THE NUMBER OF HARDWARE TESTS
                                ;       IN THE DIAGNOSTIC.
                                ;--

                                        DISPATCH 1
002122                                  .WORD   1
002122  000001
002124                          L$DISPATCH::
002124  021326                          .WORD   T1
```

```
                                .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.
                                ;       THE ACTUAL P-TABLE BUILT AT RUNTIME IS STORED IN SUPERVISOR
                                ;       SPACE.
                                ;--

                                        BGNHW   DFPTBL
002126  000003                         .WORD   L10000-L$HW/2
002130                          L$HW::
002130                          DFPTBL::
002130  172150                          .WORD   172150                  ; CSR ADDRESS
002132  000000                          .WORD   0.                     ; LOGICAL DRIVE NUMBER
002134  000000                          .WORD   0.                     ; CUSTOMER DATA AREA

                                        ENDHW
002136                          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. THIS TABLE, UNLIKE THE HARDWARE TABLE, WILL CONTAIN
                                        ;       THE ACTUAL VALUES ENTERED BY THE OPERATOR.
                                        ;--

                                                BGNSW   SFPTBL
002136  000003                                  .WORD   L10001-L$SW/2
002140                                  L$SW::
002140                                  SFPTBL::
002140  000040                                  .WORD   32.                     ;ERROR LIMIT
002142  000000                                  .WORD   0.                      ;DATA TRANSFER LIMIT (MEGABITS)
002144  040400                                  .WORD   ^B0100000100000000      ;SINGLE BIT QUESTIONS

                                                ENDSW
002146                                  L10001:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

                                ;++
                                ;       THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
                                ;       ARE USED IN MORE THAN ONE TEST.
                                ;--

                                        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
            000140              PRI03== 140
            000100              PRI02== 100
```

GLOBAL EQUATES SECTION

```
                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                  HDE==  100000
    14
```

```
                        .SBTTL  CONTROLLER BIT DEFINITIONS

                        ;
                        ;       SA REGISTER UNIVERSAL READ BITS
                        ;

100000                  SA.ERR  = 100000                        ; ERROR INDICATOR
040000                  SA.S4   = 040000                        ; STEP 4 STATUS BIT
020000                  SA.S3   = 020000                        ; STEP 3 STATUS BIT
010000                  SA.S2   = 010000                        ; STEP 2 STATUS BIT
004000                  SA.S1   = 004000                        ; STEP 1 STATUS BIT

                        ;
                        ;       SA REGISTER ERROR STATUS BITS
                        ;

003777                  SA.ERC  = 003777                        ; ERROR CODE

                        ;
                        ;       SA REGISTER STEP 1 SEND BITS
                        ;

000177                  SA.VEC  = 000177                        ; INTERRUPT VECTOR (DIVIDED BY 4)
000200                  SA.INT  = 000200                        ; INTERRUPT ENABLE DURING INIT
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

                        ;
                        ;       SA REGISTER STEP 1 RESPONSE BITS
                        ;

002000                  SA.NV   = 002000                        ; NON SETTABLE INTERRUPT VECTOR
001000                  SA.QB   = 001000                        ; 22 BIT ADDRESS BUS
000400                  SA.DI   = 000400                        ; ENHANCED DIAGNOSTICS
000100                  SA.MP   = 000100                        ; MAPPING BIT
000040                  SA.SM   = 000040                        ; SPECIAL MODE BIT FOR KDA50-Q
                        ;         000377                        ; THESE BITS RESERVED

                        ;
                        ;       SA REGISTER STEP 2 SEND BITS
                        ;

000001                  SA.PRG  = 000001                        ; ENABLE VAX UBA PURGE INTERRUPT
                        ;         177776                        ; LOW ORDER MESSAGE RING BYTE ADDRESS

                        ;
                        ;       SA REGISTER STEP 2 RESPONSE BITS
                        ;

000007                  SA.MSE  = 000007                        ; MESSAGE RING LENGTH ECHO
000070                  SA.CME  = 000070                        ; COMMAND RING LENGTH ECHO
                        ;         000100                        ; RESERVED
```

CONTROLLER BIT DEFINITIONS

```
                        SA.STE  = 000200                  ; STEP ECHO
                        SA.CTP  = 003400                  ; CONTROLLER TYPE

                        ;
                        ;       SA REGISTER STEP 3 SEND BITS
                        ;

                        ;         077777                  ; HIGH ORDER MESSAGE RING BYTE ADDRESS
                        SA.TST  = 000000                  ; PURGE/POLL TEST DISABLED
                        ;
                        ;       SA REGISTER STEP 3 RESPONSE BITS
                        ;

                        SA.VCE  = 000177                  ; INTERRUPT VECTOR ECHO
                        SA.INE  = 000200                  ; INTERRUPT ENABLE ECHO
                        SA.NVE  = 000400                  ; VECTOR NOT PROGRAMMABLE
                        ;         003000                  ; RESERVED

                        ;
                        ;       SA REGISTER STEP 4 SEND BITS
                        ;

                        SA.GO   = 000001                  ; GO BIT TO START CONTROLLER FIRMWARE
                        SA.LFC  = 000002                  ; LAST FAILURE CODE REQUEST
                        SA.BST  = 000374                  ; BURST LEVEL

                        ;
                        ;       SA REGISTER STEP 4 RESPONSE BITS
                        ;

                        SA.MCV  = 000017                  ; CONTROLLER MICROCODE VERSION
                        SA.CNT  = 000360                  ; CONTROLLER TYPE
                        ;         003400                  ; RESERVED
```

HOST COMMUNICATION AREA DEFINIIONS

```
                                .SBTTL  HOST COMMUNICATION AREA DEFINIIONS
                                ;
                                ;       COMMAND/MESSAGE RING BIT DEFINITIONS
                                ;
100000                          RG.OWN  = 100000                        ; SET WHEN CONTROLLER OWNS RING
040000                          RG.FLG  = 040000                        ; FLAG BIT
                                ;
                                ;       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
                                ;
                                ;       OFFSETS INTO HOST COMMUNICATIONS AREA WITH ONE DESCRIPTOR TO EACH RING
                                ;       AND TWO PACKETS
                                ;
                                ;               +-----------------------------------------------+
                                ;       HC.INT  )       INTERRUPT INDICATORS                    )   4 BYTES
                                ;               )                                               )
                                ;               +-----------------------------------------------+
                                ;       HC.MSG  )       MESSAGE (RESPONSE) RING                 )   4 BYTES
                                ;       HC.MCT  )                                               )
                                ;               +-----------------------------------------------+
                                ;       HC.CMD  )       COMMAND RING                            )   4 BYTES
                                ;       HC.CCT  )                                               )
                                ;               +-----------------------------------------------+
                                ;HC.MEV & HC.CEV)       MESSAGE & COMMAND ENVELOPE              )   4 BYTES
                                ;               )                                               )
                                ;HC.MPK & HC.CPK)       MESSAGE & COMMAND PACKET                )  48 BYTES
                                ;               )                                               )
                                ;               +-----------------------------------------------+
                                ;       HC.BF1  )       BUFFER # 1 (RESPONSE TO DM PROGRAM)     )  70 BYTES
                                ;               )                                               )
                                ;               +-----------------------------------------------+
                                ;       HC.BF2  )       BUFFER # 2 (REQUEST FROM DM PROGRAM)    )  70 BYTES
                                ;               )                                               )
                                ;               +-----------------------------------------------+
                                ;
                                ;               NOTE: BYTES ARE GIVEN IN DECIMAL
                                ;
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
```

HOST COMMUNICATION AREA DEFINIIONS

```
                
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
                
000014          HC.CEV  = HC.MEV                 ; COMMAND ENVELOPE START
000020          HC.CPK  = HC.MPK                 ; COMMAND PACKET START
                
000100          HC.BF1  = HC.CPK+HC.PSZ          ; FIRST BUFFER
000206          HC.BF2  = HC.BF1+HC.BSZ          ; SECOND BUFFER
                
000314          HC.SIZ  = HC.BF2+HC.BSZ          ; TOTAL SIZE OF HOST COMMUNICATION AREA
                
```

```
                                .SBTTL  COMMAND PACKET OPCODES DEFINITIONS

                                ;++
                                ;       NOTE: END PACKET OPCODES (ALSO CALLED ENDCODES) ARE FORMED BY ADDING
                                ;       THE END PACKET FLAG TO THE COMMAND OPCODE. FOR 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
                                ;--

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

                                ;
                                ;       COMMAND MODIFIERS
                                ;

                                ;       = 020000                ; CLEAR SERIOUS EXCEPTION
040000                          MD.CMP  = 040000                ; COMPARE
100000                          MD.EXP  = 100000                ; EXPRESS REQUEST
010000                          MD.ERR  = 010000                ; FORCE ERROR
```

COMMAND PACKET OPCODES DEFINITIONS

```
                        MD.SCH  = 004000        ; SUPPRESS CACHING (HIGH SPEED)
                        MD.SCL  = 002000        ; SUPPRESS CACHING (LOW SPEED)
                        MD.SEC  = 000100        ; SUPPRESS ERROR CORRECTION
                        MD.SER  = 000400        ; SUPPRESS ERROR RECOVERY
                        MD.SSH  = 000200        ; SUPPRESS SHADOWING
                        MD.WBN  = 000100        ; WRITE-BACK (NON-VOLATILE)
                        MD.WBV  = 000400        ; WRITE BACK (VOLATILE)
                        MD.SEQ  = 000020        ; WRITE SHADOW SET ONE UNIT AT A TIME
                        MD.SPD  = 000001        ; SPIN-DOWN
                        MD.FEU  = 000001        ; FLUSH ENTIRE UNIT
                        MD.VOL  = 000002        ; VOLATILE ONLY
                        MD.NXU  = 000001        ; NEXT UNIT
                        MD.RIP  = 000001        ; ALLOW SELF DESTRUCTION
                        MD.IMF  = 000002        ; IGNORE MEDIA FORMAT ERROR
                        MD.SWP  = 000004        ; SET WRITE PROTECT
                        MD.CWB  = 000010        ; CLEAR WRITE-BACK DATA LOST
                        MD.PRI  = 000001        ; PRIMARY REPLACEMENT BLOCK

                        ;
                        ;       END PACKET FLAGS
                        ;

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

                        ;
                        ;       CONTROLLER FLAGS
                        ;

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

                        ;
                        ;       UNIT FLAGS
                        ;

                        UF.CMR  = 000001        ; COMPARE READS
                        UF.CMW  = 000002        ; COMPARE WRITES
                        UF.RPL  = 100000        ; HOST INITIATED BAD BLOCK REPLACEMENT
                        UF.INA  = 040000        ; INACTIVE SHADOW SET UNIT
                        UF.SCH  = 004000        ; SUPPRESS CACHING (HIGH SPEED)
                        UF.SCL  = 002000        ; SUPPRESS CACHING (LOW SPEED)
                        UF.WBN  = 000100        ; WRITE-BACK (NON-VOLATILE)
                        UF.WPH  = 020000        ; WRITE PROTECT (HARDWARE)
                        UF.WPS  = 001000        ; WRITE PROTECT (SOFTWARE OR VOLUME)
                        UF.576  = 000004        ; 576 BYTE SECTORS

```

```
                        .SBTTL  COMMAND PACKET OFFSETS

                        ;
                        ;       GENERIC COMMAND PACKET OFFSETS
                        ;

000000                  P.CRF   = 0.                    ; COMMAND REFERENCE NUMBER
000004                  P.UNIT  = 4.                    ; UNIT NUMBER
000010                  P.OPCD  = 8.                    ; OPCODE
000012                  P.MOD   = 10.                   ; MODIFIERS
000014                  P.BCNT  = 12.                   ; BYTE COUNT
000020                  P.BUFF  = 16.                   ; BUFFER DESCRIPTOR
000020                  P.UADR  = 16.                   ; BUS 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 ADDR. (MAINT WRITE ONLY)
        000034                  P.OVRL  = 28.                                  ; BUFFER DESCRIPTOR FOR OPERLAYS

```

```
                        .SBTTL END PACKET OFFSETS

                        ;
                        ;       GENERIC END PACKET OFFSETS
                        ;

000000                  P.CRF   = 0.                    ; COMMAND REFERENCE NUMBER
000004                  P.UNIT  = 4.                    ; UNIT NUMBER
000010                  P.OPCD  = 8.                    ; OPCODE (ALSO CALLED ENDCODE)
000011                  P.FLGS  = 9.                    ; END PACKET FLAGS
000012                  P.STS   = 10.                   ; 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.MSTI  = 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.TRKS  = 36.                   ; TRACK SIZE
000046                  P.GRPS  = 38.                   ; GROUP SIZE
000050                  P.CYLS  = 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.MSTI  = 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.UNSZ  = 36.                   ; UNIT SIZE
000050                  P.VSER  = 40.                   ; VOLUME SERIAL NUMBER

                        ;
                        ;       SET CONTROLLER CHARACTERISTICS END PACKET OFFSETS
                        ;
```

END PACKET OFFSETS

```
                                ;
                                ;       GET DUST STATUS END PACKET OFFSETS
                                ;

000014                  P.DEXT  = 12.                   ; EXTENSION - DOWNLINE LOADABLE PROGRAM
000017                  P.DFLG  = 15.                   ; FLAGS
000020                  P.DPRG  = 16.                   ; PROGRESS INDICATOR FOR REMOTE PROGRAM
000024                  P.DTMO  = 20.                   ; TIMEOUT

                                ;
                                ;       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
000400                  ST.AOL  = 400                   ; ALREADY ON-LINE

                                ;
                                ;       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
```

```
                        .SBTTL  CONTROLLER TABLE DEFINITIONS

                        ;**
                        ;       CONTROLLER TABLE DEFINITIONS
                        ;
                        ;       ONE TABLE WILL BE SET UP BY INITIALIZATION SECTION FOR EACH CONTROLLER
                        ;       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.
                        ;--

000077                  CT.UNT  = 000077                        ; LOGICAL UNIT NUMBER MASK
000777                  CT.VEC  = 000777                        ; VECTOR ADDRESS MASK
007000                  CT.BRL  = 007000                        ; BR LEVEL MASK

100000                  CT.AVL  = BIT15                         ; SET WHEN NOT AVAILABLE FOR TESTING
000100                  CT.VER  = BIT6                          ; CONTROLLER MICRO VERSION NUMBER WAS
                                                                ;       PRINTED IF SET
000020                  CT.REQ  = BIT4                          ; BUFFER HAS BEEN GIVEN TO CONTROLLER
                                                                ;       FOR REQUEST. SET WHENEVER READ
                                                                ;       DUST DATA COMMAND ISSUED.
000010                  CT.MSG  = BIT3                          ; MESSAGE RESPONSE RECEIVED. WHENEVER
                                                                ;       SET, CT.CMD IS CLEARED.
000004                  CT.CMD  = BIT2                          ; COMMAND ISSUED, WAITING FOR RESPONSE
000002                  CT.RN   = BIT1                          ; DM PROGRAM RUNNING

                        TABLE                                   ; START A TABLE DEFINITION
002146                  ITEM C.UADR     2       <BUS ADDRESS OF IP REGISTER>
002146                  ITEM C.UNIT     2       <UNIT NUMBER TO TEST>
002146                  ITEM C.VEC      2       <VECTOR ADDRESS/BR LEVEL>
002146                  ITEM C.JSR      2       <INTERRUPT SERVICE ROUTINE FOR CONTROLLER>
002146                  ITEM C.JAD      2       <THESE TWO WORDS LOADED WITH [JSR PO    CNTSRV]>
002146                  ITEM C.FLG      2       <FLAGS>
002146                  ITEM C.HCOM     2       <BEGINNING ADRS OF HOST COMM AREA IN MEMORY>
002146                  ITEM C.DR0      2       <POINTER TO DRIVE TABLES>
002146                  ITEM C.DR1      2       <IF ZERO, NO DRIVE TABLE EXISTS>
002146                  ITEM C.DR2      2
002146                  ITEM C.DR3      2
002146                  ITEM C.TO       2       <TIMEOUT COUNTER>
002146                  ITEM C.TOH      2       <  (TWO WORDS)>
002146                  ITEM C.REF      2       <COMMAND REFERENCE NUMBER>

002146                  END C.SIZE      <SIZE OF CONTROLLER TABLE IN BYTES>
```

```
                                .SBTTL  DRIVE TABLE DEFINITIONS

                                ;**
                                ;       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.
                                ;
                                ;       THE FIRST TABLE IS POINTED TO BY THE CONTENTS OF DTABS.
                                ;--

            000077              DT.UNT  = 000077                        ; LOGICAL UNIT NUMBER OF DRIVE

            100000              DT.AVL  = BIT15                         ; SET WHEN NOT AVAILABLE FOR TESTING
            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

002146                          TABLE                                   ; START A TABLE DEFINITION
002146                          ITEM D.DRV      2       <DRIVE NUMBER>
002146                          ITEM D.UNIT     2       <LOGICAL UNIT NUMBER>
002146                          ITEM D.PRM      2       <SOFTWARE QUESTION FLAGS>
002146                          ITEM D.PAT      2       <DATA PATTERN NUMBER>
002146                          ITEM D.BB       2       <BAD BLOCK COUNT>
002146                          ITEM D.BB01     4       <BAD BLOCK 1>
002146                          ITEM D.BB02     4       <          2>
002146                          ITEM D.BB03     4       <          3>
002146                          ITEM D.BB04     4       <          4>
002146                          ITEM D.BB05     4       <          5>
002146                          ITEM D.BB06     4       <          6>
002146                          ITEM D.BB07     4       <          7>
002146                          ITEM D.BB08     4       <          8>
002146                          ITEM D.BB09     4       <          9>
002146                          ITEM D.BB10     4       <         10>
002146                          ITEM D.BB11     4       <         11>
002146                          ITEM D.BB12     4       <         12>
002146                          ITEM D.BB13     4       <         13>
002146                          ITEM D.BB14     4       <         14>
002146                          ITEM D.BB15     4       <         15>
002146                          ITEM D.BB16     4       <         16>
002146                          ITEM D.BEC      2       <BEGIN/END SET COUNT>
002146                          ITEM D.BGN1     4       <BEGIN BLOCK 1>
002146                          ITEM D.END1     4       <END>
```

DRIVE TABLE DEFINITIONS

```
     59 002146                          ITEM D.BGN2     4       <BEGIN BLOCK 2>
     60 002146                          ITEM D.END2     4       <END>
     61 002146                          ITEM D.BGN3     4       <BEGIN BLOCK 3>
     62 002146                          ITEM D.END3     4       <END>
     63 002146                          ITEM D.BGN4     4       <BEGIN BLOCK 4>
     64 002146                          ITEM D.END4     4       <END>
     65 002146                          ITEM D.BCYL     4       <BEGIN CYLINDER>
     66 002146                          ITEM D.ECYL     4       <END CYLINDER>
     67 002146                          ITEM D.XFRW     2       <MEGABITS WRITTEN COUNT>
     68 002146                          ITEM D.XFRR     2       <MEGABITS READ COUNT>
     69 002146                          ITEM D.HERR     2       <HARD ERROR COUNTER>
     70 002146                          ITEM D.SERR     2       <SOFT ERROR COUNTER>
     71 002146                          ITEM D.SEEK     2       <NUMBER OF SEEKS X1000>
     72 002146                          ITEM D.ECCC     2       <ECC COUNTER>
     73 002146                          ITEM D.SERN     6       <DRIVE SERIAL NUMBER>
     74 002146                          ITEM D.SKER     2       <SEEK ERROR COUNT>
     75 002146                          ITEM D.HDAS     8.      <HDA SERIAL NUMBER>
     77
     78 002146                          END D.SIZE      <SIZE OF DRIVE TABLE IN BYTES>
     79
     80                                 ;
     81                                 ;       DM PROGRAM HEADER DEFINITIONS
     82                                 ;
     83
     84          000000                 DMTRLN  = 0                     ; OFFSET TO DOWNLINE LOAD PROGRAM SIZE
     85          000004                 DMOVRL  = 4                     ; OFFSET TO SIZE OF OVERLAY
     86          000040                 DMMAIN  = 40                    ; OFFSET TO FIRST WORD OF MAIN PROGRAM
     87          001000                 DMFRST  = 1000                  ; ADDRESS START OF HEADER IN DM FILE
     88
```

```
                                .SBTTL  GLOBAL DATA SECTION

                                ;++
                                ;       THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
                                ;       IN MORE THAN ONE TEST.
                                ;--

                                        ERRTBL
002146                          L$ERRTBL::
002146                          ERRTYP::        .WORD   0
002150  000000                  ERRNBR::        .WORD   0
002152  000000                  ERRMSG::        .WORD   0
002154  000000                  ERRBLK::        .WORD   0

002156                          FFREE:: .BLKW 1                         ; FIRST FREE WORD IN MEMORY
002160                          FSIZE:: .BLKW 1                         ; SIZE OF FREE MEMORY IN WORDS
002162                          FMEM:   .BLKW 1                         ; COPY OF FFREE AT END OF INIT SECTION
002164                          FMEMS:  .BLKW 1                         ; COPY OF FSIZE AT END OF INIT SECTION
002166                          DTABS:: .BLKW 1                         ; START OF DRIVE TABLE STORAGE
002170                          CTABS:: .BLKW 1                         ; START OF CONTROLLER TABLE STORAGE
002172                          CTRLRS: .BLKW 1                         ; COUNT OF CONTROLLERS IN PTABLES
002174                          TSTTAB: .BLKW 1                         ; POINTER TO 1ST CONTROLLER TABLE
002176                          DMPROG: .BLKW 1                         ; START ADDRESS OF DM PROGRAM

002200                          IFLAGS::.BLKW 1                         ; FLAGS FROM INIT CODE

        000002                  ICONT   == BIT1                         ; CONTINUE EVENT FLAG
        000004                  IREST   == BIT2                         ; RESTART FLAG
        000010                  ISTRT   == BIT3                         ; START FLAG
        000020                  ISTRTH  == BIT4                         ; START FLAG HOLD FOR DMRQ4 ROUTINE

002202  000000                  TNUM:   .WORD 0                         ; NUMBER OF TEST EXECUTING
002204                          URUN:   .BLKW 1                         ; NUMBER OF UNITS TO RUN AT ONE TIME
002206                          URNING: .BLKW 1                         ; NUMBER OF UNITS STILL RUNNING
002210                          UCNT:   .BLKW 1                         ; COUNTER OF UNITS UNDER TEST
002212                          INTRCV: .BLKW 1                         ; INTERRUPT RECEIVED FLAG

002214                          TEMP:   .BLKW 12.                       ; TEMPORY STORAGE FOR GMANI RESPONSES

002244  000000                  IPADRS: .WORD   0                       ; 4 ENTRIES
002246  000000                          .WORD   0                       ;       FOR CONTROLLER
002250  000000                          .WORD   0                       ;               CSR
002252  000000                          .WORD   0                       ;                   ADDRESSES

002254                          BRSAV:  .BLKW 1                         ; STORE COMPUTED BR LEVEL AND VECTOR
002256  000001                  PAT16C: .WORD 1                         ; COUNT OF WORDS IN DATA PATTERN 16
002260  000000                  PAT16W: .WORD 0                         ; WORD SEQUENCE FOR DATA PATTERN 16
002262  000000                          .WORD 0
002264  000000                          .WORD 0
002266  000000                          .WORD 0
002270  000000                          .WORD 0
002272  000000                          .WORD 0
002274  000000                          .WORD 0
002276  000000                          .WORD 0
002300  000000                          .WORD 0
002302  000000                          .WORD 0
002304  000000                          .WORD 0
```

GLOBAL DATA SECTION

```
     54 002306  000000                        .WORD 0
     55 002310  000000                        .WORD 0
     56 002312  000000                        .WORD 0
     57 002314  000000                        .WORD 0
     58 002316  000000                        .WORD 0
     60
     61                                 ;
     62                                 ;       KW11 CLOCK CONTROL
     63                                 ;
     64
     65 002320  000000                  KW.CSR: .WORD 0                 ; CSR OF CLOCK
     66 002322                          KW.BRL: .BLKW 1                 ; BR LEVEL
     67 002324                          KW.VEC: .BLKW 1                 ; VECTOR
     68 002326                          KW.HZ:  .BLKW 1                 ; HERTZ (50. OR 60.)
     69 002330                          KW.EL:  .BLKW 2                 ; ELAPSED TIME
     70 002334                          STIME:  .BLKW 2                 ; STATISTICAL REPORT TIMER
     71
     72 002340                          NXMAD:  .BLKW 1                 ; SET TO -1 BY NON-EXISTANT ADDRESS
     73
     79
     80 002342  007452                  PTYPE:  .WORD   PF              ; PRINT TYPE
     81 002344     000                  TTYOUT: .BYTE   0               ; TTY OUTPUT BUFFER
     82 002345     000                          .BYTE   0               ; TERMINATOR FOR ASCIZ STRING
     83                                         .EVEN
     84
     85                                 ;
     86                                 ;       DATA TO BE SENT AND RECEIVED BY CONTROLLER INITIALIZATION
     87                                 ;
     88
     89 002346  015710                  INITBL: .WORD   RSP.S1           ; 1ST WORD RESPONSE CHECK ROUTINE
     90 002350  000000                  SND.S1: .WORD   0                ; 1ST WORD TO SEND TO SA REGISTER
     91 002352  015722                          .WORD   RSP.S2           ; 2ND WORD RESPONSE CHECK ROUTINE
     92 002354  000000                  SND.S2: .WORD   0                ; 2ND WORD TO SEND TO SA REGISTER
     93 002356  015742                          .WORD   RSP.S3           ; 3RD WORD RESPONSE CHECK ROUTINE
     94 002360  000000                  SND.S3: .WORD   0                ; 3RD WORD TO SEND TO SA REGISTER
     95                                 ;;      .WORD   RSP.S4           ; 4TH WORD RESPONSE CHECK ROUTINE
     96
     97 002362  000000                  SSTEP4: .WORD   0                ; LOCATION TO SAVE STEP 4 VALUE
     98 002364  000000                  CNTRSD: .WORD   0                ; LOCATION FOR STEP BIT MASK
    107
    109                                 ;
    110                                 ;       ERROR LOG CONTROL WORDS
    111                                 ;
    112
    113 002366                          LBUFS:  .BLKW 1                 ; START ADDRESS OF LOG/ZERO IF NONE
    114 002370                          LBUFN:  .BLKW 1                 ; ADDRESS FOR MORE DATA FOR LOG
    115 002372                          LBUFE:  .BLKW 1                 ; LAST ADDRESS AVAILABLE FOR LOG DATA
    116          001060                 LBSIZ   =       560.            ; LENGTH IN WORDS OF ERROR LOG
```

GLOBAL TEXT SECTION

```
                                .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
                                ;

                                        DEVTYP <LOGICAL DISK DRIVE>
002374                          L$DVTYP::
002374  114     117     107             .ASCIZ  /LOGICAL DISK DRIVE/
                                        .EVEN

                                ;
                                ;       TEST DESCRIPTION
                                ;

                                        DESCRIPT <CZUDIAO UDA50A,KDA50-Q DRIVE EXER>
002420                          L$DESC::
002420  103     132     125             .ASCIZ  /CZUDIAO UDA50A,KDA50-Q DRIVE EXER/
                                        .EVEN

                                ;
                                ;       UNFORMATTED MESSAGES
                                ;

002462  040     040     000     T40PT7: .ASCIZ \  \
002465  101     122     105     INITWC: .ASCIZ \ARE YOU SURE CUSTOMER DATA CAN BE DESTROYED\

                                ;
                                ;       FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

002541  045     124     000     FRMTT:  .ASCIZ \%T\
002544  045     116     000     CRLF:   .ASCIZ \%N\
002547  042     040     040     RNTIM:  .ASCIZ \"  RUNTIME "D16":"\
002572  104     071     042     RNTIM1: .ASCIZ \D9":"\
002600  104     071     000     RNTIM2: .ASCIZ \D9\
002603  042     040     040     ERRME1: .ASCIZ \"  *  *  *  ERROR PROCESSING MESSAGE STRING  *  *  *"N\
002672  116     042     122     MXFERP: .ASCIZ \N"REACHED TRANSFER LIMIT - TESTING STOPPED"N\
002747  116     042     125     ERRLIM: .ASCIZ \N"UNIT "D6" REACHED ERROR LIMIT - UNIT DROPPED FROM TEST"N\
003042  116     042     103     LOGM1:  .ASCIZ \N"CONTENTS OF ERROR LOG:"\
003074  116     042     105     LOGM2:  .ASCIZ \N"END OF ERROR LOG"N\
003121  116     042     105     LOGM3:  .ASCIZ \N"ERROR LOG IS EMPTY"N\
003150  116     042     115     T4WARN: .ASCIZ \N"MANUAL INTERVENTION NOT ALLOWED. USING DEFAULT PARAMETERS"N\
003246  116     042     103     INITWA: .ASCIZ \N"CUSTOMER DATA WILL BE DESTROYED ON:"NS5"UNIT"S3"CONTROLLER"S3"DRIVE"N\
003356  045     123     066     INITWB: .ASCIZ \%S6%D2%S6%O6%S6%D3%N\
003403  116     042     125     MESSG:  .ASCIZ \N"UNIT "D6" CONTROLLER AT "O16" DRIVE "D9S\
003456  042     116     117     NOCLOCK:.ASCIZ \"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\

003533  042     110     117     BASN0:  .ASCIZ \"HOST PROGRAM"\
003552  042     104     111     BASN4:  .ASCIZ \"DISK EXERCISER"\
003573  042     040     040     BASL1:  .ASCIZ \"  DM PC:"012\
```

GLOBAL TEXT SECTION

```
     79 003611      042     040     040  BASL2:  .ASCIZ \"  CONTROLLER AT "O16\
     80 003637      042     040     040  BASL3:  .ASCIZ \"  DRIVE "D9\
     81 003654      000                  BAS:    .BYTE 0                         ; NULL TO PRINT NOTHING
     82
     83 003655      122     066     122  BASLN:  .ASCIZ \R6R6R6R6\               ; FORMAT FOR BASIC LINE OF ERROR MESSAGE
     84
     85 003666                           X1A:
     86 003666                           X2A:
     87 003666                           X3A:
     88 003666      042     111     116  X8A:    .ASCIZ \"INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS"N\
     92 003751      122     065     122  X2:     .ASCIZ \R5R6"TWO UNITS SELECT THE SAME DRIVE"N\
     93 004020      122     065     122  X3:     .ASCIZ \R5R6"MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER"N\
     94 004106      122     064     042  X4:     .ASCII \R4"NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED"N\
     95 004167      042     122     105          .ASCIZ \"RESTART PROGRAM AND TEST FEWER UNITS AT A TIME"N\
     96 004251      122     064     042  X6:     .ASCIZ \R4"TABLE CONSISTANCY ERROR. PLEASE RE-LOAD PROGRAM"N\
    100 004336      122     065     042  X14:    .ASCII \R5"CONTROLLER IS NOT SUPPORTED BY THIS DIAGNOSTIC PROGRAM."N\
    101 004432      042     124     110          .ASCII \"THIS PROGRAM REQUIRES A UDA50A (MODEL 6, MICROCODE VERSION"\
    102 004526      042     040     101          .ASCII \" AT LEAST 3),"N\
    103 004546      042     117     122          .ASCII \"OR A KDA50-Q (MODEL 13, MICROCODE VERSION AT LEAST 0)"N\
    104 004636      116     042     103          .ASCIZ \N"CONTROLLER REPORTED MODEL CODE "D4" AND MICROCODE VERSION "D4N\
    127 004737      122     065     042  X31:    .ASCII \R5"NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES"N\
    128 005027      042     101     123          .ASCIZ \"ASSUME PROGRAM IS HUNG"N\
    129 005061      122     065     042  X32:    .ASCIZ \R5"MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER"N\
    130 005172      122     065     042  X35:    .ASCIZ \R5"DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE"N\
    131 005252      122     065     042  X36:    .ASCII \R5"NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS"N\
    132 005343      042     127     110          .ASCIZ \"WHILE LOADING DM PROGRAM"N\
    133 005377      122     065     042  X38:    .ASCII \R5"MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS"N\
    134 005464      042     103     110          .ASCII \"CHECK BUS SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE"N\
    135 005562      042     117     122          .ASCII \"OR BUS"N\
    136 005573      042     117     122          .ASCIZ \"OR "R7\
    143 005603      122     065     042  X40:    .ASCII \R5"FATAL CONTROLLER ERROR"NR8\
    144 005640      042     125     123          .ASCIZ \"USE DIAGNOSTIC PROGRAM CZUDHO TO DIAGNOSE FAILURE"N\
    146 005725      122     065     042  XFCE:   .ASCIZ \R5"FATAL ERROR REPORTED BY CONTROLLER"NR8\
    147 005777      042     115     105  XMSG1:  .ASCIZ \"MESSAGE BUFFER CONTAINS:"N\
    148 006033      123     063     117  XMSG2:  .ASCIZ \S3016S1016S1016S1016S1016S1016S1016N\
    149 006100      122     065     042  XPKT1:  .ASCII \R5"RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA"N\
    150 006203      042     105     111          .ASCII \"EITHER CONTROLLER RETURNED ERROR STATUS OR PACKET WAS RECEIVED INCORRECTLY"
N\
    151 006320      123     063     042          .ASCIZ \S3"COMMAND PACKET SENT"S6"RESPONSE PACKET RECEIVED"N\
    152 006405      123     066     117  XPKT2:  .ASCIZ \S6016S1016S14016S1016N\
    153 006434      042     123     101  XSA:    .ASCIZ \"SA REGISTER CONTAINS: "O16N\
    154 006471      042     122     105  XFRU:   .ASCIZ \"REPLACE CONTROLLER PROCESSOR MODULE"N\
    156                                          .EVEN
    157
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ;       THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS USED
                                ;       BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PNTB
                                ;       (BASIC) AND PNTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                                ;--


006540                          BGNMSG ERR002
006540                          ERR002::
006540                                  PNTB    X2,@X2A
006554                          ENDMSG
006554                          L10002:
006554  104423                          TRAP    C$MSG

006556                          BGNMSG ERR003
006556                          ERR003::
006556                                  PNTB    X3,@X3A
006572                          ENDMSG
006572                          L10003:
006572  104423                          TRAP    C$MSG

006574                          BGNMSG ERR004
006574                          ERR004::
006574                                  PNTB    X4
006604                          ENDMSG
006604                          L10004:
006604  104423                          TRAP    C$MSG

006606                          BGNMSG ERR006
006606                          ERR006::
006606                                  PNTB    X6
006616                          ENDMSG
006616                          L10005:
006616  104423                          TRAP    C$MSG

006620                          BGNMSG ERR014
006620                          ERR014::
006620                                  PNTB    X14,R3,R1
006634                          ENDMSG
006634                          L10006:
006634  104423                          TRAP    C$MSG

006636                          BGNMSG ERR030
006636                          ERR030::
006636                                  PNTB    XFCE,R1
006650                          ENDMSG
006650                          L10007:
006650  104423                          TRAP    C$MSG

006652                          BGNMSG ERR031
006652                          ERR031::
006652                                  PNTB    X31
006662                          ENDMSG
```

```
                006662
                006662  104423                  L10010:
                                                        TRAP    C$MSG
    107
    108         006664                          BGNMSG ERR032
                006664                          ERR032::
    109         006664                                  PNTB    X32
    110         006674  004737  007060                  CALL    MSGPKT
    111         006700                          ENDMSG
                006700                          L10011:
                006700  104423                          TRAP    C$MSG
    112
    113         006702                          BGNMSG ERR033
                006702                          ERR033::
    114         006702  004737  006766                  CALL    PNTPKT
    115         006706                          ENDMSG
                006706                          L10012:
                006706  104423                          TRAP    C$MSG
    116
    117         006710                          BGNMSG ERR035
                006710                          ERR035::
    118         006710                                  PNTB    X35
    119         006720  004737  007060                  CALL    MSGPKT
    120         006724                          ENDMSG
                006724                          L10013:
                006724  104423                          TRAP    C$MSG
    121
    122         006726                          BGNMSG ERR036
                006726                          ERR036::
    123         006726                                  PNTB    X36
    124         006736                          ENDMSG
                006736                          L10014:
                006736  104423                          TRAP    C$MSG
    125
    126         006740                          BGNMSG ERR038
                006740                          ERR038::
    127         006740                                  PNTB    X38
    128         006750                          ENDMSG
                006750                          L10015:
                006750  104423                          TRAP    C$MSG
    129
    136         006752                          BGNMSG ERR040
                006752                          ERR040::
    137         006752                                  PNTB    X40,R2
    138         006764                          ENDMSG
                006764                          L10016:
                006764  104423                          TRAP    C$MSG
    140
    141         006766                          PNTPKT: PNTB    XPKT1
    142         006776  010401                          MOV     R4,R1
    143         007000  062701  000020                  ADD     #HC.CPK,R1
    144         007004  010402                          MOV     R4,R2
    145         007006  062702  000020                  ADD     #HC.MPK,R2
    146         007012  012703  000014                  MOV     #12.,R3
    147
    148         007016                          PNTPKL: PNTB    XPKT2,2(R1),(R1),2(R2),(R2)
    149         007042  062701  000004                  ADD     #4,R1
    150         007046  062702  000004                  ADD     #4,R2
```

```
                                        DEC     R3
                                        BNE     PNTPKL
                                        RETURN

                                MSGPKT: PNTB    XMSG1
                                        MOV     C.HCOM(R5),R4
                                        ADD     #HC.BF2,R4
                                        MOV     #5,R3
                                MSGPKL: PNTB    XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)
                                        ADD     #14.,R4
                                        DEC     R3
                                        BNE     MSGPKL
                                        RETURN

                                BGNMSG ERR.TN                           ; ERROR REPORT ROUTINE
                                ERR.TN::
                                        MOV     TNUM,R2                 ; GET TEST NUMBER
                                        ASL     R2                      ; DOUBLE
                                        MOV     #BASL3,R3               ; GET ADDRESS OF DRIVE PRINT LINE
                                        TST     4(R4)                   ; CHECK IF DRIVE NUMBER GIVEN
                                        BPL     1$                      ;       BRANCH IF SO
                                        MOV     #BAS,R3
                                1$:     PNTB    BASLN,TNAMES-2(R2),#BASL1,(R4),#BASL2,(R5),R3,4(R4)
                                        ASSUME C.UADR EQ 0
                                        CALL    RNTIME                  ; GET RUNTIME PARAMETERS
                                        PRINT   #CR                     ; ADVANCE TO NEW LINE
                                        ADD     #6,R4                   ; SET R4 TO POINT TO MESSAGE POINTER
                                        MOV     (R4)+,R2                ; GET MESSAGE POINTER
                                        ASL     R2                      ; DOUBLE TO MAKE BYTE OFFSET
                                        ADD     DMPROG,R2               ; ADD TO START OF MESSAGE STRINGS
                                        ADD     #DMPROG,R2              ; ADD SIZE OF MAIN PROGRAM
                                        TSTB    (R2)                    ; CHECK FIRST BYTE
                                        BNE     NCON                    ; IF ZERO
                                        INC     R2                      ;       INCREMENT TO NEXT BYTE
                                NCON:   MOV     #PX,PTYPE               ; CHANGE TO EXTENDED OUTPUT
                                        CALL    OSTRNG                  ; OUTPUT ACCORDING TO STRING
                                ENDMSG
                                L10017:
                                        TRAP    C$MSG
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;**
                                ;       FMERR - MEMORY ALLOCATION ERROR
                                ;
                                ;       THIS ROUTINE PRINTS A SYSTEM FATAL ERROR AND EXITS THE TEST
                                ;--

007320                          FMERR:  ERRSF 4,,ERR004
007320  104454                         TRAP    C$ERSF
007322  000004                         .WORD   4
007324  000000                         .WORD   0
007326  006574                         .WORD   ERR004
                                                                ; DO CLEAN-UP TRAP
007330                                  DOCLN
007330  104444                         TRAP    C$DCLN

                                ;**
                                ;       ALOCM - ALLOCATE A BLOCK OF FREE MEMORY.  REPORT ERROR IF MEMORY
                                ;       EXHAUSTED.
                                ;
                                ;       INPUTS:
                                ;               R1 - NUMBER OF WORDS TO ALLOCATE
                                ;               FFREE - FIRST FREE WORD IN MEMORY
                                ;               FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS
                                ;       OUTPUTS:
                                ;               R1 - ADDRESS OF FIRST WORD OF ALLOCATED MEMORY
                                ;               FFREE - NEW FIRST FREE WORD IN MEMORY
                                ;               FSIZE - SIZE OF FREE MEMORY LEFT AFTER ALLOCATION
                                ;
                                ;       SYSTEM FATAL ERROR WILL BE REPORTED IF NOT ENOUGH MEMORY AVAILABLE
                                ;       AND ENTIRE PROGRAM WILL BE STOPPED.
                                ;--

007332                          ALOCM:  PUSH    <FFREE>                 ; SAVE FFREE AT ENTRY
007336  160137  002160                  SUB     R1,FSIZE                ; REDUCE SIZE OF FREE MEMORY
007342  002766                         BLT     FMERR                   ; REPORT ERROR IF NOT ENOUGH MEMORY
007344  060101                         ADD     R1,R1                   ; CHANGE WORDS TO BYTES
007346  060137  002156                 ADD     R1,FFREE                ; CALCULATE NEW START OF FREE MEMORY
007352                                  POP     <R1>                    ; GET START OF ALLOCATED MEMORY
007354  000207                         RETURN

                                ;**
                                ;       HCOMM - ALLOCATE MEMORY FOR HOST COMMUNICATION AREA AND PACKET
                                ;       BUFFERS WITH ONE DESCRIPTOR IN EACH RING. THIS SUBROUTINE IS 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 COMMUNICATION AREA.
                                ;               RING POINTERS TO PACKETS.
                                ;               R4 - ADDRESS OF HOST COMMUNICATION AREA
                                ;--

007356  012701  000146          HCOMM:  MOV     @<HC.SIZ>/2,R1          ; GET SIZE OF AREA TO ALLOCATE
007362  004737  007332                  CALL    ALOCM                   ; ALLOCATE THE MEMORY
```

```
     007366  010104                          MOV     R1,R4                   ; GET ADDRESS OF MOST COMM AREA
     007370  010465  000014                  MOV     R4,C.HCOM(R5)           ; PLACE BEGINNING ADRS OF MOST COMM
                                                                             ;       AREA IN THE CONTROLLER TABLE
     007374  062701  000020                  ADD     #HC.MPK,R1              ; COMPUTE START OF MESSAGE PACKET
     007400  010164  000004                  MOV     R1,HC.MSG(R4)           ; PLACE IN RING
     007404  062701  000000                  ADD     #<HC.CPK-HC.MPK>,R1     ; COMPUTE START OF COMMAND PACKET
     007410  010164  000010                  MOV     R1,HC.CMD(R4)           ; PLACE IN RING
     007414  000207                          RETURN

                                     ;**
                                     ;       PRINTC - PRINT A CHARACTER
                                     ;
                                     ;       CALL WITH MACRO PRINT
                                     ;--

     007416  110037  002344          PRINTC: MOVB    R0,TTYOUT               ; SAVE CHARACTER FOR TTY OUTPUT
     007422                          PUSH    <R1>
     007424  012701  002541          MOV     #FRMTT,R1               ; GET FORMATTED ASCIZ STRING STATEMENT
     007430  120027  000015          CMPB    R0,#CR                  ; IF NOT A CARRIAGE RETURN, THEN
     007434  001002                  BNE     1$                      ; PRINT SOME OTHER CHARACTER, ELSE
     007436  012701  002544          MOV     #CRLF,R1                ; GET FORMATTED ASCIZ STRING STATEMENT
                                                                     ;       GO PRINT CR-LF.
     007442  004777  172674  1$:     CALL    @PTYPE                  ; PRINT THE ASCIZ STRING.
     007446                          POP     <R1>
     007450  000207                  RETURN
     007452                  PF:     PRINTF  R1,#TTYOUT
     007452  012746  002344          MOV     #TTYOUT,-(SP)
     007456  010146                  MOV     R1,-(SP)
     007460  012746  000002          MOV     #2,-(SP)
     007464  010600                  MOV     SP,R0
     007466  104417                  TRAP    C$PNTF
     007470  062706  000006          ADD     #6,SP
     007474  000207                  RETURN
     007476                  PB:     PRINTB  R1,#TTYOUT
     007476  012746  002344          MOV     #TTYOUT,-(SP)
     007502  010146                  MOV     R1,-(SP)
     007504  012746  000002          MOV     #2,-(SP)
     007510  010600                  MOV     SP,R0
     007512  104414                  TRAP    C$PNTB
     007514  062706  000006          ADD     #6,SP
     007520  000207                  RETURN
     007522                  PX:     PRINTX  R1,#TTYOUT
     007522  012746  002344          MOV     #TTYOUT,-(SP)
     007526  010146                  MOV     R1,-(SP)
     007530  012746  000002          MOV     #2,-(SP)
     007534  010600                  MOV     SP,R0
     007536  104415                  TRAP    C$PNTX
     007540  062706  000006          ADD     #6,SP
     007544  000207                  RETURN
     007546                  PS:     PRINTS  R1,#TTYOUT
     007546  012746  002344          MOV     #TTYOUT,-(SP)
     007552  010146                  MOV     R1,-(SP)
     007554  012746  000002          MOV     #2,-(SP)
     007560  010600                  MOV     SP,R0
     007562  104416                  TRAP    C$PNTS
     007564  062706  000006          ADD     #6,SP
     007570  000207                  RETURN
```

```
                                        ;++
                                        ;       PRINT FORMATTED MESSAGE
                                        ;
                                        ;       CALL WITH MACRO PNT, PNTF, PNTB, PNTX, OR PNTS
                                        ;--

007572  012737  007452  002342  LPNTF:  MOV     #PF,PTYPE
007600  000413                          BR      LPNT

007602  012737  007476  002342  LPNTB:  MOV     #PB,PTYPE
007610  000407                          BR      LPNT

007612  012737  007522  002342  LPNTX:  MOV     #PX,PTYPE
007620  000403                          BR      LPNT

007622  012737  007546  002342  LPNTS:  MOV     #PS,PTYPE

007630                          LPNT:   PUSH    <R2,R3,R4,R5>
007640  012102                          MOV     (R1)+,R2                ; GET ADDRESS OF ASCIZ STRING
007642  010604                          MOV     SP,R4                   ; COMPUTE ADDRESS OF 1ST ARGUMENT AND
007644  062704  000012                  ADD     #12,R4                  ; SAVE IT IN R4.
007650                                  PUSH    <R1>                    ; SAVE R1 ON THE STACK
007652  004737  007676                  CALL    OSTRNG                  ; PRINT THE FORMATTED MESSAGE
007656                                  POP     <R0,R5,R4,R3,R2,R1>     ; RESTORE REGISTERS FROM STACK
007672  062006                          ADD     (R0)+,SP                ; ADJUST STACK POINTER OVER ARGUMENTS
007674  000110                          JMP     @R0                     ; 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
                                ;--

007676  112201                  OSTRNG: MOVB    (R2)+,R1                ; SEE IF TERMINATOR IN ASCIZ STRING,
007700  001421                          BEQ     OSTRE                   ; EXIT
007702  012700  010202                  MOV     #ERRC,R0                ; GET POINTER TO CHARACTER TABLE
007706  120110                  NCONS:  CMPB    R1,(R0)                 ; COMPARE CHARACTER WITH TABLE ENTRY
007710  001407                          BEQ     NCONF                   ; BRANCH IF MATCH FOUND
007712  105720                          TSTB    (R0)+                   ; INCREMENT POINTER
007714  001374                          BNE     NCONS                   ; CONTINUE SEARCH IF NOT END OF TABLE
007716                                  PNTF    ERRME1                  ; REPORT BAD CONTROL CHARACTER
007726  000406                          BR      OSTRE                   ;
007730  162700  010202          NCONF:  SUB     #ERRC,R0                ; GET INCREMENT INTO TABLE
007734  006300                          ASL     R0                      ; DOUBLE TO WORD COUNT
007736  004770  010214                  CALL    @ERRD(R0)               ; DISPATCH TO PRINT ROUTINE
007742  000755                          BR      OSTRNG                  ; GET NEXT
007744  000207                  OSTRE:  RETURN

                                ;
                                ;       CONTROL CHARACTER WAS A QUOTE, SO  PRINT ALL CHARACTERS TO
                                ;       THE NEXT QUOTE.
                                ;

007746  112200                  CON.QU: MOVB    (R2)+,R0                ; GET CHARACTER
007750  120027  000042                  CMPB    R0,#'"                  ; CHECK IF ENDING QUOTE
007754  001403                          BEQ     CON.QX                  ; IF SO, GO GET NEXT CONTROL CHARACTER
007756                                  PRINT   R0
007762  000771                          BR      CON.QU                  ;       CONTINUE PRINTING
007764  000207                  CON.QX: RETURN

                                ;
                                ;       CONTROL CHARACTER WAS AN 'A', SO PRINT ASCII CHARACTERS FROM
                                ;       PARAMETERS.
                                ;

007766  004737  013314          CON.A:  CALL    GETCNT                  ; GET COUNT OF CHARACTERS
007772                          CON.A1: PRINT   (R4)+                   ;
010000  005301                          DEC     R1                      ; COUNT THE CHARACTERS
010002  001373                          BNE     CON.A1                  ; PRINT UNTIL COUNT REACHES ZERO
010004  032704  000001                  BIT     #1,R4                   ; CHECK IF R4 NOW ODD
010010  001401                          BEQ     CON.A2                  ;
010012  005204                          INC     R4                      ; IF SO, INCREMENT TO NEXT EVEN ADDRESS
010014  000207                  CON.A2: RETURN                          ; NOW GET NEXT CONTROL CHARACTER

                                ;
                                ; CONTROL CHARACTER WAS A 'D', SO PRINT A DECIMAL NUMBER.
                                ;

010016  012701  000012          CON.D:  MOV     #10.,R1                 ; LOAD RADIX
010022  004737  013372                  CALL    PNTNUM                  ; PRINT NUMBER
010026  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER

                                ;
                                ;       CONTROL CHARACTER WAS AN 'H', SO PRINT A HEX NUMBER.
```

```
                                        ;
                                        ;
010030  012701  000020          CON.H:  MOV     #16.,R1                 ; LOAD RADIX
010034  004737  013372                  CALL    PNTNUM                  ; PRINT NUMBER
010040  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'O', SO PRINT AN OCTAL NUMBER.
                                        ;
010042  012701  000010          CON.O:  MOV     #8.,R1                  ; LOAD RADIX
010046  004737  013372                  CALL    PNTNUM                  ; PRINT NUMBER
010052  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'N', SO PRINT A CARRIAGE RETURN-LINE FEED.
                                        ;
010054  004737  013314          CON.N:  CALL    GETCNT                  ; GET COUNT
010060                          CON.N1: PRINT   #CR                     ;
010070  005301                          DEC     R1                      ; COUNT THE SEQUENCES
010072  001372                          BNE     CON.N1                  ;
010074  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'R', SO CALL ONE OF THE PRE-PROGRAMMED
                                        ;       ROUTINE.
                                        ;
010076  004737  013314          CON.R:  CALL    GETCNT                  ; GET ROUTINE NUMBER
010102  020127  000011                  CMP     R1,#ERR.SZ              ; CHECK IF DEFINED ROUTINE NUMBER
010106  101004                          BHI     CON.R1                  ;
010110  060101                          ADD     R1,R1                   ; DOUBLE COUNT TO GET WORD INDEX
010112  004771  010154                  CALL    @ERR.TB-2(R1)           ; CALL ROUTINE
010116  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
010120                          CON.R1: PNTF    ERRME1                  ; REPORT BAD MESSAGE STRING
010130                                  POP     R1                      ; FIX THE STACK
010132  000207                          RETURN
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'S', SO PRINT SOME NUMBER OF SPACES.
                                        ;
010134  004737  013314          CON.S:  CALL    GETCNT                  ; GET COUNT
010140                          CON.S1: PRINT   <#' >                   ;
010150  005301                          DEC     R1                      ; COUNT THE SPACES
010152  001372                          BNE     CON.S1                  ;
010154  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       PRE-PROGRAMMED ERROR ROUTINE DISPATCH TABLE
                                        ;
010156  012656                  ERR.TB: .WORD   CALR1                   ; CALL ALTERNATE PRINT STRING IN DM PGM
010160  012704                          .WORD   CALR2                   ; PRINT AN SDI DIAGNOSE RESPONSE
010162  013002                          .WORD   CALR3                   ; DECIDE WHETHER TO PRINT RBN
```

```
                                        .WORD CALR4                     ; PRINT BASIC LINE W/O CONTROLLER ADDR.
                                        .WORD CALR5                     ; PRINT BASIC LINE W/ CONTROLLER ADDR.
                                        .WORD CALR6                     ; CALL ALTERNATE PRINT STRING IN PDP-11
                                        .WORD CALR7                     ; PRINT "REPLACE PROCESSOR MODULE"
                                        .WORD CALR8                     ; PRINT "SA REGISTER CONTAINS  XXXXXX"
                                        .WORD CALR9                     ; REPRINT LAST NUMBER

000011                                  ERR.SZ  = <.-ERR.TB>/2

                                TNAMES:
                                        .WORD BASN4

                                ;
                                ;       CONTROL CHARACTER TABLE
                                ;

                                ERRC:   .BYTE   '"
                                        .BYTE   'A
                                        .BYTE   'D
                                        .BYTE   'H
                                        .BYTE   'O
                                        .BYTE   'N
                                        .BYTE   'R
                                        .BYTE   'S

                                        .BYTE   0                       ;FOLLOW WITH A NULL BYTE
                                        .EVEN

                                ;
                                ;       ROUTINE ADDRESS TABLE
                                ;

                                ERRD:   .WORD   CON.QU
                                        .WORD   CON.A
                                        .WORD   CON.D
                                        .WORD   CON.H
                                        .WORD   CON.C
                                        .WORD   CON.N
                                        .WORD   CON.R
                                        .WORD   CON.S
```

```
                                        ;**
                                        ;
                                        ;       TINIT - INITIALIZE VARIABLES FOR TEST
                                        ;
                                        ;       INPUTS:
                                        ;               R1 - TEST NUMBER
                                        ;       OUTPUTS:
                                        ;               DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
                                        ;               LBUFS - CLEARED (DELETES ERROR LOG)
                                        ;               TNUM - TEST NUMBER FROM R1
                                        ;               ALL REGISTERS CLOBERED
                                        ;--

010234                                  TINIT:
010234  010137  002202                          MOV     R1,TNUM                 ; SAVE TEST NUMBER
010240  004737  016134                          CALL    RESET                   ; RESET ALL CONTROLLERS
010244  005037  002366                          CLR     LBUFS                   ; CLEAR ERROR LOG BUFFER POINTER
010250  013737  002162  002156                  MOV     FMEM,FFREE              ; INIT FREE
010256  013737  002164  002160                  MOV     FMEMS,FSIZE             ; INIT FSIZE
010264  006301                                  ASL     R1                      ; R1 IS WORD INDEX
010266  016137  010274  002176                  MOV     READDT-2(R1),DMPROG     ; STORE ADDRESS OF DM TEST INTO DMPROG
010274  000207                                  RETURN

010276                                  READDT:
010276  000000G                               .WORD   TEST4
```

GLOBAL SUBROUTINES SECTION

```
                                        ;++
                                        ;       RUNDM - LOAD AND RUN A DM PROGRAM IN THE CONTROLLERS. RETURN WHEN ALL
                                        ;               DM PROGRAMS HAVE TERMINATED.
                                        ;
                                        ;       INPUTS:
                                        ;               R1 - NUMBER OF CONTROLLERS TO TEST
                                        ;               R5 - POINTER TO FIRST CONTROLLER TABLE
                                        ;       IMPLICIT INPUTS:
                                        ;               DMPROG - POINTER TO START OF DM PROGRAM IN MEMORY
                                        ;       OUTPUTS:
                                        ;               Z SET IF NO CONTROLLERS SUCCESSFULLY STARTED
                                        ;               R1-R5 - CONTENTS DESTROYED
                                        ;--

010300  010137  002204          RUNDM:  MOV     R1,URUN                 ; SAVE NUMBER OF UNITS TO RUN
010304  005037  002206                  CLR     URNING                  ; CLEAR NUMBER OF UNITS RUNNING
                                        ;
                                        ;       LOAD DM PROGRAM INTO EACH CONTROLLER
                                        ;
010310  013705  002174                  MOV     TSTTAB,R5               ; GET CONTROLLER TABLE ADDRESS
010314  013737  002204  002210          MOV     URUN,UCNT               ; SET COUNTER OF UNITS
010322                          LDDM:
010322  005065  000012                  CLR     C.FLG(R5)               ; CLEAR ALL FLAGS
010326  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; SEE IF UNIT TO BE TESTED
010334  005765  000002                  TST     C.UNIT(R5)              ;
010340  100405                          BMI     LDNEXT                  ; IF NOT, DON'T LOAD THIS UNIT
010342                                  ASSUME CT.AVL EQ BIT15
010342  004737  014376                  CALL    LOADDM                  ; LOAD THE DM PROGRAM
010346  001402                          BEQ     LDNEXT                  ; IF ERROR, GO TO NEXT CONTROLLER
010350  005237  002206                  INC     URNING                  ; IF NO ERROR, COUNT UNIT RUNNING
010354  062705  000034          LDNEXT: ADD     #C.SIZE,R5              ; MOVE TO NEXT CONTROLLER TABLE
010360  005337  002210                  DEC     UCNT                    ; CHECK IF MORE CONTROLLERS
010364  001356                          BNE     LDDM                    ; LOAD NEXT CONTROLLER
                                        ;
                                        ;       CHECK IF ANY CONTROLLERS LOADED
                                        ;
010366  005737  002206                  TST     URNING                  ; ANY UNITS LOADED?
                                        ;
                                        ;       THE DM PROGRAMS ARE NOW IN CONTROL
                                        ;       RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
                                        ;
010372  000207                          RETURN                          ;
```

```
                                        ;++
                                        ;       RESPDM - RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
                                        ;                HAVE TERMINATED.
                                        ;
                                        ;       INPUTS:
                                        ;                R5 - CONTROLLER TABLE ADDRESS
                                        ;--
                                        RESPDM: MOV     TSTTAB,R5               ; GET CONTROLLER TABLE ADDRESS
                                                MOV     URUN,UCNT               ; SET COUNTER OF UNITS
                                        RESPCT: MOV     C.HCOM(R5),R4           ; GET HOST COMM AREA ADDRESS
                                                BIT     #CT.RN,C.FLG(R5)        ; CHECK IF PROGRAM RUNNING
                                                BEQ     RSPNXT                  ;       IF NOT, LOOK AT NEXT
                                                MOVB    C.UNIT(R5),L$LUN        ; STORE UNIT NUMBER UNDER TEST
                                                BIT     #CT.MSG,C.FLG(R5)       ; SEE IF INTERRUPT RECEIVED
                                                BNE     RSPIN                   ;       IF SO, LOOK AT PACKET
                                                BIT     #CT.CMD,C.FLG(R5)       ; SEE IF COMMAND HAS BEEN SENT
                                                BEQ     RSPOU                   ;       IF NOT, SEND ONE
                                        ;
                                        ;       CHECK IF CONTROLLER STILL RUNNING
                                        ;
                                                MOV     (R5),R3                 ; GET ADDRESS OF IP REGISTER
                                                MOV     2(R3),R1                ; LOOK AT SA REGISTER
                                                BEQ     RSPTM                   ; IF ZERO, CONTROLLER STILL RUNNING
                                                ERRDF 30,,ERR030                ; ELSE, REPORT FATAL CONTROLLER ERROR
                                                TRAP    C$ERDF
                                                .WORD   30
                                                .WORD   0
                                                .WORD   ERR030
                                                BR      RSPDRP                  ; DROP CONTROLLER FROM TESTING
                                        ;
                                        ;       CHECK FOR TIMEOUT OF RESPONSE
                                        ;
                                        RSPTM:
                                                TST     KW.CSR                  ; SEE IF A CLOCK ON SYSTEM
                                                BEQ     RSPNTO                  ; DON'T TIME IF NO CLOCK
                                                CMP     KW.EL+2,C.TOH(R5)       ; CHECK HIGH WORD OF ELAPSED TIME
                                                BHI     RSPTMO                  ; IF GREATER, RESPONSE TIMED OUT
                                                BNE     RSPNTO                  ; IF NOT SAME, ITS OK
                                                CMP     KW.EL,C.TO(R5)          ; CHECK LOW WORD OF ELAPSED TIME
                                                BLO     RSPNTO                  ; IF LESS, PLENTY OF TIME LEFT
                                        RSPTMO: ERRDF 31,,ERR031                ; REPORT TIMEOUT ERROR
                                                TRAP    C$ERDF
                                                .WORD   31
                                                .WORD   0
                                                .WORD   ERR031
                                                BR      RSPDRP                  ; DROP CONTROLLER FROM TESTING
                                        RSPNTO:
                                                BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
                                                TRAP    C$BRK
                                        ;
                                        ;       CHECK FOR TIME TO PRINT STATISTICAL REPORT
                                        ;
                                        RSPNXT: TST     KW.CSR                  ; ANY CLOCK ON SYSTEM?
                                                BEQ     RSPNRP                  ;       BYPASS IF NOT
                                                CMP     KW.EL+2,STIME+2         ; CHECK HIGH WORD OF ELAPSED TIME
```

```
                                                BHI     RSPRPT                  ; IF GREATER PRINT REPORT
                                                BNE     RSPNRP                  ; IF NOT SAME, ITS NOT TIME YET
                                                CMP     KW.EL,STIME             ; CHECK LOW WORD OF ELAPSED TIME
                                                BLO     RSPNRP                  ; IF LESS, ITS NOT TIME YET
                                        RSPRPT:                                 ;
                                                DORPT                           ; PRINT A STATISTICAL REPORT
                                                TRAP    C$DRPT
                                        ;
                                        ;       SWITCH TO NEXT CONTROLLER
                                        ;
                                        RSPNRP: ADD     #C.SIZE,R5              ; MOVE TO NEXT TABLE
                                                DEC     UCNT                    ; CHECK IF MORE CONTROLLERS
                                                BNE     RESPCT                  ; LOOK AT NEXT CONTROLLER
                                                BR      RESPDM                  ; LOOK AT FIRST CONTROLLER AGAIN
                                        ;
                                        ;       REMOVE A CONTROLLER FROM TESTING
                                        ;
                                        RSPDRP: BIC     #CT.RN+CT.MSG,C.FLG(R5) ; CLEAR PROGRAM RUNNING
                                                DEC     URNING                  ; REDUCE RUNNING CONTROLLERS COUNT
                                                BNE     RSPNXT                  ; IF ANY STILL RUNNING, LOOK AT THEM
                                                RETURN                          ; ELSE RETURN TO TEST SECTION
                                        ;
                                        ;       CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
                                        ;       CHECK FOR PROPER OPCODE IN END PACKET
                                        ;
                                        RSPIN:  MOV     #OP.END+OP.SSD,R0       ; GET SEND DATA END PACKET OPCODE
                                                BIT     #CT.REQ,C.FLG(R5)       ; LOOK IF SEND DATA OR RECEIVE DATA
                                                BEQ     RSPMMR                  ;
                                                MOV     #OP.END+OP.RSD,R0       ; CHANGE TO RECEIVE DATA ENDCODE
                                        RSPMMR: CMPB    R0,HC.MPK+P.OPCD(R4)    ; COMPARE TO OPCODE IN END PACKET
                                                BNE     RSPERR                  ;
                                        ;
                                        ;       LOOK AT STATUS CODE
                                        ;
                                                BIT     #ST.MSK,HC.MPK+P.STS(R4); CHECK FOR STATUS CODE ST.SUC (ZERO)
                                                BNE     RSPERR                  ;
                                        ;
                                        ;       CHECK FOR EXPECTED REFERENCE NUMBER
                                        ;
                                                CMP     C.REF(R5),HC.MPK+P.CRF(R4); CHECK IF CORRECT REF NUMBER
                                                BEQ     RSPPTW                  ;
                                        RSPERR: ERRDF 33,,ERR033                ;
                                                TRAP    C$ERDF
                                                .WORD   33
                                                .WORD   0
                                                .WORD   ERR033
                                                BR      RSPDRP                  ; DROP UNIT FROM TESTING
                                        ;
                                        ;       CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
                                        ;
                                        RSPPTW: BIT     #CT.REQ,C.FLG(R5)       ; CHECK IF RESPONSE FROM DM PROGRAM
                                        RSPOU:  BEQ     RSPOUT                  ; LOOK AT REQUEST NUMBER IF SO
                                        ;
                                        ;       MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
                                        ;
                                        RSPPT2: MOV     HC.BF2(R4),R1           ; GET REQUEST NUMBER
                                                BIC     #007777,R1              ; CHECK TYPE
```

```
                                        CMP     #DU.SPC,R1              ; IS SPECIAL TYPE SET?
                                        BNE     1$                      ; IF NOT, ERROR
                                        BIC     #^C007777,HC.BF2(R4)    ; CLEAR TYPE
                                        MOV     HC.BF2(R4),R1           ; GET REQUEST NUMBER
                                        CMP     R1,#DSPSIZ              ; CHECK IF IN EXPECTED RANGE
                                        BLO     RSPPT3                  ;
                                1$:     ERRDF 32.,ERR032                ; BAD REQUEST NUMBER
                                        TRAP    C$ERDF
                                        .WORD   32
                                        .WORD   0
                                        .WORD   ERR032
                                        BR      RSPDRP                  ; DROP UNIT FROM TESTING

                                RSPPT3: MOV     #OP.SSD,R0              ; BUILD A SEND DATA COMMAND PACKET
                                        CALL    BLDCMD                  ;    FOR ANSWER TO DM PROGRAM
                                        MOV     #HC.BF1,R0              ; POINT TO BUFFER IN PACKET
                                        CALL    CLRBUF                  ;    AND CLEAR BUFFER
                                        MOV     R4,R3                   ; R3 POINTS TO COMMAND BUFFER
                                        ADD     #HC.BSZ,R4              ; R4 POINTS TO MESSAGE BUFFER
                                        MOV     (R4),R1                 ; GET REQUEST NUMBER
                                        MOV     (R4)+,(R3)+             ; PUT REQUEST # INTO COMMAND PACKET
                                        ADD     R1,R1                   ; DOUBLE REQUEST NUMBER
                                        CALL    @RSPDSP(R1)             ; CALL REQUESTED ROUTINE
                                        BNE     RSPDRP                  ; RETURN Z CLEAR TO DROP UNIT
                                                                        ; Z SET IF READY TO SEND COMMAND

                                ;
                                ;       SEND COMMAND BACK TO CONTROLLER
                                ;
                                RSPOUT: BIC     #CT.MSG,C.FLG(R5)       ; CLEAR MESSAGE RECEIVED FLAG
                                        BIT     #CT.REQ,C.FLG(R5)       ; CHECK WHICH COMMAND TO SEND
                                        BNE     RSPOU2                  ; BRANCH IF RESPONSE TO REQUEST
                                        MOV     #OP.RSD,R0              ; BUILD RECEIVE DATA COMMAND
                                        CALL    BLDCMD                  ;
                                        MOV     #HC.BF2,R0              ; POINT TO MESSAGE BUFFER
                                        CALL    CLRBUF                  ;    AND CLEAR IT
                                        BIS     #CT.REQ,C.FLG(R5)       ; SET REQUEST BIT
                                        BR      RSPOU3                  ;

                                RSPOU2: BIC     #CT.REQ,C.FLG(R5)       ; CLEAR REQUEST BIT
                                RSPOU3:
                                        CALL    SNDCMD                  ; SEND COMMAND TO CONTROLLER
                                        MOV     #3.*60.,R0              ; SET TIMEOUT FOR 3 MINUTES
                                        MOV     R5,R1                   ;
                                        ADD     #C.TO,R1                ; PUT TIME IN CONTROLLER TABLE
                                        CALL    SETTO                   ;
                                        JMP     RSPNXT                  ; NOW WAIT FOR END PACKET

                                ;
                                ;       RESPONSE REQUEST DISPATCH TABLE
                                ;
                                RSPDSP: .WORD DMRQ0                     ; 0 SET UP MEMORY FOR ADDRESS TESTING
                                        .WORD 0                         ; 1 *** REQUEST NO LONGER VALID ***
                                        .WORD 0                         ; 2 *** NOT VALID FOR CZUDIO ***
                                        .WORD DMRQ3                     ; 3 TELL DATA PATTERN 16.
                                        .WORD DMRQ4                     ; 4 TELL UNIT PARAMS, CLEAR CONTENTS
                                        .WORD DMRQ5                     ; 5 TELL BAD BLOCKS (FIRST 14)
                                        .WORD DMRQ6                     ; 6 TELL BAD BLOCKS (LAST TWO)
                                        .WORD DMRQ7                     ; 7 ADD TO SOFT ERROR AND ECC COUNTS
```

```
174 011146  011734                 .WORD DMRQ8                 ; 8 ADD 1000 TO SEEK COUNT
175 011150  011754                 .WORD DMRQ9                 ; 9 ADD TO MEGABITS TRANSFERRED COUNT
176 011152  012116                 .WORD DMRQA                 ;10 TELL WHICH DRIVES TO TEST
177 011154  012176                 .WORD DMRQB                 ;11 REPORT ERROR MESSAGE
178 011156  012402                 .WORD DMRQC                 ;12 REPORT ERROR & COUNT HARD ERROR
179 011160  012540                 .WORD DMRQD                 ;13 PRINT A DESCRIPTIVE MESSAGE
180 011162  012652                 .WORD DMRQE                 ;14 MARK DM PROGRAM AS NOT RUNNING
182
183         000017                 DSPSIZ = <.-RSPDSP>/2       ;LEGAL NUMBERS ARE LOWER THAN THIS
184
```

```
                        ;++
                        ;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        )
                        ;       .                       .
                        ;       .                       .
                        ;       .                       .
                        ;       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
                        ;       6       )       DATA ARGUMENT #3        )  CONTAINING ZEROS UNLESS
                        ;       .                       .                  SPECIFICALLY INDICATED BY
                        ;       .                       .                  RESPONSE ROUTINE.
                        ;       .                       .
                        ;       68      )       DATA ARGUMENT #34       )
                        ;
```

```
                        ;++
                        ;       DMRQ0  -  DM REQUEST 0
                        ;
                        ;       SET UP MEMORY FOR ADDRESS TESTING FROM CONTROLLER.
                        ;       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:
                        ;               0.(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
                        ;--
011164                  DMRQ0:
011164  013701  002156          MOV     FFREE,R1                ;GET FIRST ADDRESS OF FREE MEMORY
011170  013702  002160          MOV     FSIZE,R2                ;GET SIZE
                        ;
                        ;       FILL MEMORY WITH ADDRESS PATTERN
                        ;
011174  010111          MEMFIL: MOV     R1,(R1)                 ;WRITE DATA INTO LOCATION
011176  062701  000002          ADD     #2,R1                   ;INCREASE ADDRESS TO NEXT LOCATION
011202  005302                  DEC     R2                      ;COUNT THE WORDS
011204  001373                  BNE     MEMFIL                  ;FILL ALL WORDS
                        ;
                        ;       SEND LOCATION OF FREE MEMORY TO CONTROLLER
                        ;
011206  013723  002156          MOV     FFREE,(R3)+             ;LOAD FIRST ADDRESS OF FREE MEMORY
011212  005023                  CLR     (R3)+                   ; HIGH ORDER BITS ARE ZERO
011214  013700  002160          MOV     FSIZE,R0                ;GET SIZE OF FREE MEMORY
011220  006300                  ASL     R0                      ;CONVERT TO BYTES
011222  063700  002156          ADD     FFREE,R0                ;COMPUTE LAST LOCATION
011226  162700  000002          SUB     #2,R0
011232  010023                  MOV     R0,(R3)+                ;LOAD LAST LOCATION
011234  005023                  CLR     (R3)+                   ;CLEAR HIGH ORDER BITS
                        ;
                        ;       SEND LOCATION OF READABLE MEMORY
                        ;
011236  005023                  CLR     (R3)+                   ;SEND ZERO AS START OF READABLE MEMORY
011240  005023                  CLR     (R3)+
011242  013700  0C2120          MOV     L#HIMEM,R0              ;GET HIGH MEMORY ADDRESS
011246  005001                  CLR     R1                      ;CLEAR HIGH BITS
011250  006300                  ASL     R0                      ;SHIFT LEFT 6 PLACES
```

```
 58 011252  006300                          ASL     R0
 59 011254  006300                          ASL     R0
 60 011256  006300                          ASL     R0
 61 011260  006300                          ASL     R0
 62 011262  006101                          ROL     R1
 63 011264  006300                          ASL     R0
 64 011266  006101                          ROL     R1
 65 011270  052700  000076                  BIS     #76,R0                  ;SET LOW ORDER BITS
 66 011274  010023                          MOV     R0,(R3)+                ;PUT INTO BUFFER
 67 011276  010123                          MOV     R1,(R3)+
 68 011300  000264                          SEZ
 69 011302  000207                          RETURN
 70
```

```
 1
 2                                      ;DMRQ3 - DM REQUEST 3
 3                                      ;
 4                                      ;REQUEST FOR CZUDIO CONTENTS OF DATA PATTERN 16.
 5                                      ;
 6                                      ;INPUTS:
 7                                      ;       R5 - CONTROLLER TABLE ADDRESS
 8                                      ;       R4 - MESSAGE DATA ADDRESS
 9                                      ;            (NO DATA)
10                                      ;       R3 - COMMAND DATA ADDRESS
11                                      ;OUTPUTS:
12                                      ;       COMMAND DATA FILLED WITH THE FOLLOWING:
13                                      ;          (R3) NUMBER OF WORDS IN DATA PATTERN 16
14                                      ;           2.(R3) DATA IN PATTERN 16
15                                      ;             :             "
16                                      ;          32.(R3)          "
17                                      ;          Z SET
18
19 011304  012701  000021              DMRQ3:  MOV     #17.,R1                 ;GET COUNT
20 011310  012702  002256                      MOV     #PAT16C,R2              ; AND ADDRESS OF PATTERN 16 PARAMETERS
21 011314  012223                      1$:     MOV     (R2)+,(R3)+             ;COPY THE DATA TO BUFFER
22 011316  005301                              DEC     R1
23 011320  001375                              BNE     1$
24 011322  000264                              SEZ                             ;RETURN WITH Z SET
25 011324  000207                              RETURN
```

```
                        ;DMRQ4 - DM REQUEST 4
                        ;
                        ;REQUEST FOR CZUDIO 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
                        ;
011326                  DMRQ4:
```

```
                                                    MOV     (R4)+,R1                ;GET DRIVE NUMBER
                                                    MOV     R4,R2                   ;SAVE DATA ADDRESS
                                                    CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
                                                    BNE     DMRQ4X                  ;CHECK IF DRIVE FOUND
                                                    MOV     (R2)+,D.SERN(R4)        ;COPY DRIVE SERIAL # TO DRIVE TABLE
                                                    MOV     (R2)+,D.SERN+2(R4)
                                                    MOV     (R2)+,D.SERN+4(R4)
                                                    MOV     (R2)+,D.MDAS(R4)        ;COPY MDA SERIAL NUMBER TO DRIVE TABLE
                                                    MOV     (R2)+,D.MDAS+2(R4)
                                                    MOV     (R2)+,D.MDAS+4(R4)
                                                    MOV     D.PRM(R4),R1            ;GET PARAMETER BITS
                                                    BIC     #D.ZERO,R1              ;CLEAR SOME BITS
                                                    BIT     #ISTRTH,IFLAGS          ;FIRST TIME CZUDIO BEING RUN,
                                                    BEQ     1$                      ;BRANCH IF NOT, ELSE
                                                    BIT     #SM.IW,SFPTBL+SO.BIT    ;GET INITIAL WRITE BIT.
                                                    BEQ     1$
                                                    BIS     #D.IW,R1                ;MOVE INTO PARAMETER BITS
                                                1$: MOV     R1,(R3)+                ;PUT INTO BUFFER
                                                    MOV     D.PAT(R4),(R3)+         ;PUT PATTERN NUMBER IN BUFFER
                                                    BIT     #D.BE,R1                ;CHECK BEGIN/END PARAMETER BIT
                                                    BEQ     3$                      ;BRANCH IF NOT SET
                                                ;
                                                ;   RETURN BEGIN/END SETS
                                                ;
                                                    MOV     #4*4+1,R1               ; # OF SETS * WORDS PER SET+COUNT WORD
                                                    MOV     R4,R2                   ;GET INDEX INTO DRIVE TABLE
                                                    ADD     #D.BEC,R2
                                                2$: MOV     (R2)+,(R3)+             ;TRANSFER THE BEGIN/END SETS
                                                    DEC     R1
                                                    BNE     2$
                                                    BR      DMRQ4X

                                                3$: BIT     #D.CYL,D.PRM(R4)        ;LOOK AT D.CYL BIT
                                                    BEQ     8$                      ;BRANCH IF NOT SET

                                                ;
                                                ;   RETURN TRACKS/GROUPS AND CYLINDERS
                                                ;

                                                    TST     D.BEC(R4)               ;CHECK IF ANY TRACKS/GROUPS
                                                    BEQ     6$                      ;BRANCH IF NONE
                                                    MOV     #4,R1                   ;COUNT OF CYLINDER WORDS
                                                    MOV     R4,R2
                                                    ADD     #D.BCYL,R2
                                                4$: MOV     (R2)+,(R3)+             ;CYLINDERS
                                                    DEC     R1
                                                    BNE     4$
                                                    MOV     #8.,R1
                                                    MOV     R4,R2
                                                    ADD     #D.BEC,R2
                                                5$: MOV     (R2)+,(R3)+             ;TRACKS/GROUPS
                                                    DEC     R1
                                                    BNE     5$
                                                    BR      DMRQ4X
                                                ;
                                                ;   RETURN CYLINDERS ONLY
                                                ;
```

GLOBAL SUBROUTINES SECTION

```
    115 011542  052763  000040  177774  6$:     BIS     #D.BE,-4(R3)            ;SET D.BE FOR DM PROGRAM
    116 011550  005023                          CLR     (R3)+                   ;SEND ZERO BEGIN/END COUNT
    117 011552  012701  000004                  MOV     #4,R1
    118 011556  010402                          MOV     R4,R2
    119 011560  062702  000154                  ADD     #D.BCYL,R2
    120 011564  012223                  7$:     MOV     (R2)+,(R3)+             ;CYLINDERS
    121 011566  005301                          DEC     R1
    122 011570  001375                          BNE     7$
    123 011572  000412                          BR      DMRQ4X
    124
    125                                 ;
    126                                 ;       RETURN ENTIRE AREA
    127                                 ;
    128
    129 011574  052763  000040  177774  8$:     BIS     #D.BE,-4(R3)            ;SET D.BE FOR DM PROGRAM
    130 011602  005023                          CLR     (R3)+                   ;BEGIN/END COUNT OF ZERO
    131 011604  005023                          CLR     (R3)+                   ;START CYLINDER OF ZERO
    132 011606  005023                          CLR     (R3)+
    133 011610  005023                          CLR     (R3)+                   ;END CYLINDER NEGATIVE
    134 011612  012723  177777                  MOV     #-1,(R3)+
    135 011616  000264                          SEZ
    136 011620  000207                  DMRQ4X: RETURN
```

```
                                        ;DMRQ5 - 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

011622  011401                  DMRQ5:  MOV     (R4),R1                 ;GET DRIVE NUMBER
011624  004737  013224                  CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
011630  001007                          BNE     DMRQ5E                  ;CHECK IF DRIVE FOUND
011632  062704  000010                  ADD     #D.BB,R4                ;INCREASE ADDRESS TO DATA TO COPY
011636  012701  000035                  MOV     #<1+<14.*2>>,R1         ;GET COUNT OF WORDS
011642  012423                  1$:     MOV     (R4)+,(R3)+             ;COPY THE WORDS
011644  005301                          DEC     R1
011646  001375                          BNE     1$
011650  000207                  DMRQ5E: RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;DMRQ6 - 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
                                        
011652  011401                  DMRQ6:  MOV     (R4),R1                 ;GET DRIVE NUMBER
011654  004737  013224                  CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
011660  001007                          BNE     DMRQ6E                  ;CHECK IF DRIVE FOUND
011662  062704  000102                  ADD     #D.BB15,R4              ;INCREASE ADDRESS TO DATA TO COPY
011666  012701  000004                  MOV     #4,R1                   ;GET COUNT OF WORDS
011672  012423                  1$:     MOV     (R4)+,(R3)+             ;COPY THE WORDS
011674  005301                          DEC     R1
011676  001375                          BNE     1$
011700  000207                  DMRQ6E: RETURN
```

```
                                        ;++
                                        ;       DMRQ7 - DM REQUEST 7
                                        ;
                                        ;       CZUDIO - 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
                                        ;                6.(R4) VALUE TO ADD TO SEEK ERROR COUNT
                                        ;               R3 - COMMAND DATA ADDRESS
                                        ;--

011702  012401                  DMRQ7:  MOV     (R4)+,R1                ;GET DRIVE NUMBER
011704  010402                          MOV     R4,R2                   ;SAVE DATA ADDRESS
011706  004737  013224                  CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
011712  001007                          BNE     1$                      ;CHECK IF DRIVE FOUND
011714  062264  000172                  ADD     (R2)+,D.SERR(R4)        ;ADD TO SOFT ERROR COUNT
011720  062264  000176                  ADD     (R2)+,D.ECCC(R4)        ;ADD TO ECC COUNT
011724  062264  000206                  ADD     (R2)+,D.SKER(R4)        ;ADD TO SEEK ERROR COUNT
011730  000264                          SEZ                             ;EXIT
011732  000207                  1$:     RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;++
                                        ;       DMRQ8 - DM REQUEST 8.
                                        ;
                                        ;       RECORD 1000 SEEKS COMPLETED ON DRIVE
                                        ;
                                        ;       INPUTS:
                                        ;               R5 - CONTROLLER TABLE ADDRESS
                                        ;               R4 - MESSAGE DATA ADDRESS
                                        ;                  (R4) DRIVE NUMBER
                                        ;               R3 - COMMAND DATA ADDRESS
                                        ;
                                        ;--
011734  011401                  DMRQ8:  MOV     (R4),R1                 ; GET DRIVE NUMBER
011736  004737  013224                  CALL    GTDRVT                  ; GET DRIVE TABLE ADDRESS
011742  001003                          BNE     SEKERE                  ; CHECK IF DRIVE FOUND
011744  005264  000174                  INC     D.SEEK(R4)              ; COUNT THE BITS TRANSFERRED
011750  000264                          SEZ                             ; NORMAL RETURN
011752  000207                  SEKERE: RETURN
```

```
 1
 2                                      ;DMRQ9 - DM REQUEST 9.
 3                                      ;
 4                                      ;RECORD 1M BITS TRANSFERRED ON UNIT. COMPARE TO TRANSFER LIMIT AND
 5                                      ;REPORT LIMIT REACHED.
 6                                      ;
 7                                      ;INPUTS:
 8                                      ;       R5 - CONTROLLER TABLE ADDRESS
 9                                      ;       R4 - MESSAGE DATA ADDRESS
10                                      ;          (R4) DRIVE NUMBER
11                                      ;           2.(R4) VALUE TO ADD TO READ COUNT
12                                      ;           4.(R4) VALUE TO ADD TO WRITE COUNT
13                                      ;       R3 - COMMAND DATA ADDRESS
14                                      ;OUTPUTS:
15                                      ;          (R3) BIT 15 SET IF TRANSFER LIMIT REACHED
16                                      ;       MESSAGE PRINTED IF TRANSFER LIMIT REACHED
17                                      ;       Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
18
19 011754  010402                      DMRQ9:  MOV     R4,R2                   ;GET MESSAGE DATA ADDRESS
20 011756  011401                              MOV     (R4),R1                 ;GET DRIVE NUMBER
21 011760  004737  013224                      CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
22 011764  001053                              BNE     MXFERE                  ;CHECK IF DRIVE FOUND
23 011766  005764  000002                      TST     D.UNIT(R4)              ;SEE IF UNIT HAS BEEN DROPPED
24 011772  100003                              BPL     1$                      ;CONTINUE IF STILL TO BE TESTED
25 011774                                      ASSUME DT.AVL EQ BIT15
26 011774  052713  100000                      BIS     #BIT15,(R3)             ;TELL DM PROGRAM TO STOP TESTING THIS UNIT
27 012000  000444                              BR      MXFERX                  ;  AND EXIT WITHOUT ADDING TO ADDING TO COUNTS
28 012002                              1$:
29 012002  066264  000002  000166              ADD     2(R2),D.XFRR(R4)        ;ADD MEGABITS READ
30 012010  066264  000004  000164              ADD     4(R2),D.XFRW(R4)        ;ADD MEGABITS WRITTEN
31 012016  005737  002142                      TST     SFPTBL+SO.XL            ;SEE IF LIMIT SPECIFIED
32 012022  001433                              BEQ     MXFERX                  ;BRANCH IF NOT
33 012024  026437  000166  002142              CMP     D.XFRR(R4),SFPTBL+SO.XL ;CHECK IF LIMIT REACHED
34 012032  103427                              BLO     MXFERX                  ;BRANCH IF LIMIT NOT REACHED
35 012034                                      RFLAGS  R0                      ;CHECK FLAGS
   012034  104421                              TRAP    C$RFLA
36 012036  032700  000040                      BIT     #IDU,R0                 ;SEE IF DROPPING UNITS IS INHIBITED
37 012042  001023                              BNE     MXFERX
38 012044  052713  100000                      BIS     #BIT15,(R3)             ;SET DROP UNIT BIT
39 012050  042765  000010  000012              BIC     #CT.MSG,C.FLG(R5)          ;CLEAR MESSAGE RECEIVED FLAG
40 012056                                      PNTX    MESSG,D.UNIT(R4),(R5),(R4) ;PRINT TESTING DONE
41 012076  004737  016236                      CALL    RNTIME                     ;PRINT RUNTIME
42 012102                                      PNTX    MXFERP
43 012112  000264                      MXFERX: SEZ                             ;NORMAL RETURN
44 012114  000207                      MXFERE: RETURN
```

```
                                ;DMRQA - 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 4 DRIVE NUMBERS.
                                ;         LIST IS ENDED BY A WORD WITH BIT 15 SET.
                                ;       D.XFRW, D.XFRR, D.HERR, D.SERR, D.SEEK AND D.ECC CLEARED
                                ;       Z SET
                                
012116  010504          DMRQA:  MOV     R5,R4                   ;GET ADDRESS OF CONTROLLER TABLE
012120  062704  000016          ADD     #C.DRV,R4               ;BUMP TO DRIVE TABLE POINTERS
012124  012702  000004          MOV     #4.,R2                  ;GET COUNT OF PORTS
012130  012400          UTOT1:  MOV     (R4)+,R0                ;SEE IF DRIVE TABLE POINTER EXISTS
012132  001415                  BEQ     UTOT2                   ;BRANCH IF NOT
012134  005760  000002          TST     D.UNIT(R0)              ;LOOK IF UNIT AVAILABLE FOR TESTING
012140  100410                  BMI     UTOT1A
012142                          ASSUME DT.AVL EQ BIT15
012142  011023                  MOV     (R0),(R3)+              ;LOAD DRIVE NUMBER FROM TABLE
012144  062700  000164          ADD     #D.XFRW,R0              ;CLEAR STATISTICS IN DRIVE TABLE
012150  012701  000016          MOV     #<D.SIZE-D.XFRW>/2,R1
012154  005020          1$:     CLR     (R0)+
012156  005301                  DEC     R1
012160  001375                  BNE     1$
012162  005302          UTOT1A: DEC     R2                      ; COUNT THE DRIVE TABLES
012164  001361                  BNE     UTOT1                   ; REPEAT FOR EACH TABLE
012166  012723  100000  UTOT2:  MOV     #BIT15,(R3)+            ; TERMINATE LIST
012172  000264                  SEZ                             ;
012174  000207                  RETURN                          ; RETURN WITH Z SET
```

```
                                ;DMRQB  - 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
012176                          DMRQB:
                                ;;      PUSH    R4                      ; SAVE R4
                                ;;      MOV     4(R4),R1                ; R1 = DRIVE #
                                ;;      BMI     1$                      ; IF -1, THEN NO DRIVE # GIVEN
                                ;;      CALL    GTDRVT                  ; GET DRIVE TABLE ADDRESS
                                ;;      TST     D.UNIT(R4)              ; IF DRIVE HAS BEEN DROPPED, EXIT
                                ;;      BGE     1$                      ; NEGATIVE VALUE MEANS DRIVE DROPPED
                                ;;      POP     R4                      ;
                                ;;      BIS     #BIT15,(R3)             ; SET DROP DRIVE BIT
                                ;;      BR      8$                      ; EXIT
                                ;;1$:   POP     R4                      ;RESTORE R4
                                ;;
012176  005764  000002                  TST     2(R4)                   ;CHECK IF FATAL ERROR
012202  100406                          BMI     5$                      ;BRANCH IF NOT
012204                                  RFLAGS  R0                      ;LOOK AT FLAGS
012204  104421                          TRAP    C$RFLA
012206  032700  000040                  BIT     #IDU,R0                 ;SEE IF ALLOWED TO DROP UNITS
012212  001014                          BNE     6$                      ;BRANCH IF NOT
012214  052713  100000                  BIS     #BIT15,(R3)             ;SET DROP DRIVE BIT
012220  016400  000002          5$:     MOV     2(R4),R0                ;SEE IF SOFT ERROR
012224  005100                          COM     R0
012226  032700  140000                  BIT     #140000,R0
012232  001004                          BNE     6$                      ;BRANCH IF NOT
012234  032737  000400  002144          BIT     #SM.SSF,SO.BIT+SFPTBL   ;SEE IF SOFT ERRORS SUPPRESSED
012242  001055                          BNE     ERRMSX                  ;DON'T PRINT IF SO
012244                          6$:
012244  042765  000010  000012          BIC     #CT.MSG,C.FLG(R5)       ;CLEAR MESSAGE RECEIVED FLAG
012252  032737  001000  002144          BIT     #SM.LOG,SFPTBL+SO.BIT   ; SEE IF LOG BEING USED
012260  001005                          BNE     ERRMSL                  ; IF SO, LOG IT ELSE
012262  004737  014232                  CALL    PNTERR                  ; PRINT ERROR MESSAGE
012266  103043                          BCC     ERRMSX                  ; IF DRIVE HASN'T BEEN DROPPED, PRINT
012270                          8$:
```

GLOBAL SUBROUTINES SECTION

```
     59 012270  000244                          CLZ                             ;       ELSE RETURN
     60 012272  000207                          RETURN
     61
     63 012274  005737  002366          ERRMSL: TST     LBUFS                   ; SEE IF ERROR LOG BUFFER ESTABLISHED
     64 012300  001014                          BNE     1$                      ; ZERO INDICATES BUFFER NOT CREATED
     65 012302  012701  001060                  MOV     #LBSIZ,R1               ; GET SIZE OF BUFFER TO CREATE
     66 012306  004737  007332                  CALL    ALOCM                   ; ALLOCATE MEMORY FOR BUFFER
     67 012312  010137  002366                  MOV     R1,LBUFS                ; GET POINTER TO TOP OF BUFFER
     68 012316  010137  002370                  MOV     R1,LBUFN                ; POINT TO START OF FREE AREA
     69 012322  062701  002140                  ADD     #<LBSIZ*2>,R1           ; CALCULATE END OF ERROR LOG
     70 012326  010137  002372                  MOV     R1,LBUFE                ; SAVE END OF ERROR LOG
     71 012332  013701  002370          1$:     MOV     LBUFN,R1                ; GET ADDRESS OF DATA STORAGE AREA
     72 012336  062737  000106  002370          ADD     #HC.BSZ,LBUFN           ; ADD BYTES OF STORAGE NEEDED
     73 012344  023737  002370  002372          CMP     LBUFN,LBUFE             ; SEE IF ENOUGH ROOM
     74 012352  103007                          BHIS    3$                      ;       BRANCH IF NOT
     75 012354  010521                          MOV     R5,(R1)+                ; STORE CONTROLLER TABLE ADDRESS
     76 012356  012700  000042                  MOV     #<HC.BSZ-2>/2,R0        ; GET COUNT OF REST OF DATA IN WORDS
     77 012362  012421                  2$:     MOV     (R4)+,(R1)+             ; STORE DATA
     78 012364  005300                          DEC     R0                      ; NEXT WORD
     79 012366  001375                          BNE     2$                      ; END OF DATA?
     80 012370  000402                          BR      ERRMSX                  ; EXIT
     81 012372  010137  002370          3$:     MOV     R1,LBUFN                ; RESTORE OLD VALUE OF LBUFN
     83 012376  000264                  ERRMSX: SEZ
     84 012400  000207                          RETURN
     85
```

```
                                ;DMRQC - DM REQUEST 12.
                                ;
                                ;REPORT AN ERROR MESSAGE IDENTICAL TO DM REQUEST DMRQB
                                ;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)                 "
                                ;            :                    "
                                ;            :                    "
                                ;         38.(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

012402                          DMRQC:  PUSH    R4
012404  004737  012176                  CALL    DMRQB                   ; CALL REQUEST DMRQB
012410                                  POP     R4
012412  005713                          TST     (R3)                    ;SEE IF UNIT ALREADY TO BE DROPPED
012414  100445                          BMI     3$                      ;  IF SO, JUST EXIT NOW
012416  016401  000004                  MOV     4(R4),R1                ; GET DRIVE NUMBER
012422  016402  000002                  MOV     2(R4),R2                        ;GET ERROR TYPE
012426  004737  013224                  CALL    GTDRVT                  ; GET DRIVE TABLE
012432  001040                          BNE     5$                      ; EXIT IF NO TABLE FOR UNIT
012434  042702  037777                  BIC     #^C140000,R2
012440  022702  100000                  CMP     #100000,R2              ;CHECK IF HARD ERROR
012444  001031                          BNE     3$                      ;BRANCH IF NOT
012446  005264  000170                  INC     D.HERR(R4)              ; COUNT THE ERROR
012452  026437  000170  002140          CMP     D.HERR(R4),SFPTBL+S0.EL ; CHECK IF AT LIMIT
012460  103423                          BLO     3$                      ; IF LIMIT REACHED, BRANCH
012462                                  RFLAGS  R0                      ;LOOK AT THE FLAGS
012462  104421                          TRAP    C$RFLA
012464  032700  000040                  BIT     #IDU,R0                 ;SEE IF DROPPING UNITS INHIBITED
012470  001017                          BNE     3$                      ;BRANCH IF SO
012472  005764  000002                  TST     D.UNIT(R4)              ; ALL READY BEEN DROPPED?
012476  100003                          BPL     2$                      ; IF SO, DO NOT PRINT LINE
012500  052713  100000                  BIS     #BIT15,(R3)             ;SET STOP TESTING BIT
012504  000411                          BR      3$
012506                          2$:
012506                                  PNTX    ERRLIM,D.UNIT(R4)       ; PRINT LIMIT REACHED
012522  052713  100000                  BIS     #BIT15,(R3)             ;SET STOP TESTING BIT
                                                                        ;PRINT A STATISTICAL REPORT
012526                                  DORPT
```

GLOBAL SUBROUTINES SECTION

```
                012526  104424                          TRAP    C$DRPT
     57
     58 012530  000264                  3$:     SEZ                     ; SET Z FOR NORMAL RETURN
     59 012532  000207                          RETURN                  ; RETURN TO CALLING PROGRAM
     60
     61 012534  000244                  5$:     CLZ                     ; FLAG AS ERROR
     62 012536  000207                          RETURN                  ; RETURN TO CALLING PROGRAM
```

```
                                ;DMRQD - DM REQUEST 13.
                                ;
                                ;PRINT A MESSAGE WITH HEADER AS FOLLOWS:
                                ;   "UNIT XX CONTROLLER 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

012540  042765  000010  000012  DMRQD:  BIC     #CT.MSG,C.FLG(R5)       ;CLEAR MESSAGE RECEIVED FLAG
012546  012401                          MOV     (R4)+,R1                ;GET DRIVE NUMBER
012550                                  PUSH    R4                      ;SAVE DATA POINTER
012552  004737  013224                  CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
012556  001033                          BNE     1$                      ;CHECK IF DRIVE FOUND
012560  005764  000002                  TST     D.UNIT(R4)              ;IF UNIT DROPPED FROM TESTING
012564  100430                          BMI     1$                      ;    DON'T PRINT ANYTHING
012566                                  PNTX    MESSG,D.UNIT(R4),(R5),(R4); PRINT HEADER
012606  004737  016236                  CALL    RNTIME                  ; GET RUNTIME PARAMETERS
012612                                  POP     R4                      ; RESTORE MESSAGE MOINTER
012614  012402                          MOV     (R4)+,R2                ;GET MESSAGE POINTER
012616  006302                          ASL     R2                      ;DOUBLE TO MAKE BYTE OFFSET
012620  063702  002176                  ADD     DMPROG,R2               ;ADD TO START OF MESSAGE STRINGS
012624  063702  167346                  ADD     $DMPROG,R2              ;ADD SIZE OF MAIN PROGRAM
012630  105712                          TSTB    (R2)                    ;CHECK FIRST BYTE
012632  001001                          BNE     2$                      ;IF ZERO
012634  005202                          INC     R2                      ;   INCREMENT TO NEXT BYTE
012636  004737  007676          2$:     CALL    OSTRNG                  ;OUTPUT ACCORDING TO STRING
012642  000264                          SEZ
012644  000207                          RETURN
012646                          1$:     POP     R4
012650  000207                          RETURN
```

GLOBAL SUBROUTINES SECTION

```
                                        ;DMRQE - 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
                                        
012652  000244                        DMRQE:  CLZ                             ;DROP UNIT FROM TESTING
012654  000207                                RETURN
```

```
                                        .SBTTL  PRE-PROGRAMMED SUBROUTINES

                                        ;**
                                        ;       CALR1 - PRE-PROGRAMMED PRINT ROUTINE 1
                                        ;
                                        ;       CALL ALTERNATE PRINT STRING IN DM PROGRAM IMAGE
                                        ;--

012656                                  CALR1:  PUSH    R2
012660  012402                                  MOV     (R4)+,R2                ;GET NEW STRING POINTER
012662  006302                                  ASL     R2                      ;DOUBLE FOR WORD COUNT
012664  063702  002176                          ADD     DMPROG,R2               ;ADD START OF STRING STORAGE
012670  067702  167302                          ADD     @DMPROG,R2              ;ADD SIZE OF MAIN PROGRAM
012674  004737  007676                          CALL    OSTRNG                  ;OUTPUT USING THIS STRING
012700                                          POP     R2                      ;GET OLD POINTER BACK
012702  000207                                  RETURN                          ;NOW CONTINUE THE OLD STRING

                                        ;**
                                        ;       CALR2 - PRE-PROGRAMMED PRINT ROUTINE 2
                                        ;
                                        ;       PRINT AN SDI DIAGNOSE RESPONSE
                                        ;--

012704                                  CALR2:  PUSH    R2
012706  012402                                  MOV     (R4)+,R2                ;GET COUNTS
012710                                          PUSH    R2                      ;SAVE COUNTS
012712  042702  177400                          BIC     #177400,R2              ;GET BINARY COUNT
012716  001414                                  BEQ     2$                      ;BYPASS BINARY IF COUNT IS ZERO
012720  012700  000020                  1$:     MOV     #16.,R0                 ;RADIX IS HEX
012724  012701  000040                          MOV     #32.,R1                 ;32 BIT NUMBERS
012730  004737  013400                          CALL    PNTNUS                  ;PRINT THE NUMBER
012734                                          PRINT   #CR                     ;GO TO NEW LINE
012744  005302                                  DEC     R2
012746  001364                                  BNE     1$
012750                                  2$:     POP     R1
012752  000301                                  SWAB R1                         ;GET ASCII COUNT
012754  042701  177400                          BIC     #177400,R1
012760  001406                                  BEQ     3$                      ;BYPASS IS COUNT IS ZERO
012762  004737  007772                          CALL    CON.A1                  ;PRINT THE ASCII
012766                                          PRINT   #CR                     ;GO TO NEW LINE
012776                                  3$:     POP     R2
013000  000207                                  RETURN

                                        ;**
                                        ;       CALR3 - PRE-PROGRAMMED PRINT 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
                                        ;--

013002  032724  000200          CALR3:  BIT     #BIT7,(R4)+             ;CHECK BIT 7 IN FIRST PARAMETER WORD
013006  001323                          BNE     CALR1                   ;IF SET, TURN INTO A CALR1
```

```
 58 013010  062704  000006                ADD     #6,R4                   ;ELSE, SKIP OVER NEXT 3 PARAMETERS
 59 013014  000207                        RETURN
 60
 61                                ;++
 62                                ;       CALR4 - PRE-PROGRAMMED PRINT ROUTINE 4
 63                                ;
 64                                ;       PRINT BASIC LINE FOR HOST PROGRAM ERROR WITHOUT CONTROLLER ADDRESS
 65                                ;       THEN SWITCH TO EXTENDED FORMAT
 66                                ;--
 67
 68 013016                        CALR4:  PNTB    BASLN,#BASNO,#BAS,#BAS,#BAS
 69 013046  004737  016236                CALL    RNTIME
 70 013052                                PRINT   #CR
 71 013062  012737  007522  002342        MOV     #PX,PTYPE
 72 013070  000207                        RETURN
 73
 74                                ;++
 75                                ;       CALR5 - PRE-PROGRAMMED PRINT ROUTINE 5
 76                                ;
 77                                ;       PRINT BASIC LINE FOR HOST PROGRAM ERROR WITH CONTROLLER ADDRESS
 78                                ;       THEN SWITCH TO EXTENDED FORMAT
 79                                ;--
 80
 81 013072                        CALR5:  PNTB    BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
 82 013124  004737  016236                CALL    RNTIME
 83 013130                                PRINT   #CR
 84 013140  012737  007522  002342        MOV     #PX,PTYPE
 85 013146  000207                        RETURN
 86
 87                                ;++
 88                                ;       CALR6 - PRE-PROGRAMMED PRINT ROUTINE 6
 89                                ;
 90                                ;       CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
 91                                ;--
 92
 93 013150                        CALR6:  PUSH    R2
 94 013152  012402                        MOV     (R4)+,R2                ;GET NEW STRING POINTER
 95 013154  004737  007676                CALL    OSTRNG                  ;OUTPUT USING THIS STRING
 96 013160                                POP     R2
 97 013162  000207                        RETURN                          ;NOW CONTINUE THE OLD STRING
 98
 99                                ;++
100                                ;       CALR7 - PRE-PROGRAMMED PRINT ROUTINE 7
101                                ;
102                                ;       PRINT "REPLACE CONTROLLER PROCESSOR MODULE"
103                                ;--
104
105 013164                        CALR7:  PUSH    R2
106 013166  012702  006471                MOV     #XFRU,R2
107 013172  004737  007676                CALL    OSTRNG
108 013176                                POP     R2
109 013200  000207                        RETURN
110
111                               ;++
112                               ;       CALR8 - PRE-PROGRAMMED PRINT ROUTINE 8
113                               ;
114                               ;       PRINT  "  SA REGISTER CONTAINS  XXXXXX"
```

PRE-PROGRAMMED SUBROUTINES

```
                                        ;--
                                        ;
013202                          CALR8:  PUSH    R2
013204  012702  006434                  MOV     #XSA,R2
013210  004737  007676                  CALL    OSTRNG
013214                                  POP     R2
013216  000207                          RETURN
                                        ;
                                        ;++
                                        ;       CALR9 - PRE-PROGRAMMED PRINT ROUTINE 9
                                        ;
                                        ;       REPRINT LAST NUMBER
                                        ;               R4 -> TABLE
                                        ;--
                                        ;
013220  005744                  CALR9:  TST     -(R4)
013222  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

013224                                  GTDRVT: PUSH    R2
013226  010504                                  MOV     R5,R4                   ;GET CONTROLLER TABLE ADRESS
013230  062704  000016                          ADD     #C.DRO,R4               ;ADD OFFSET TO DRIVE TABLE ADDRESS
013234  012702  000004                          MOV     #4.,R2                  ;GET COUNT OF DRIVES
013240  005714                          1$:     TST     (R4)                    ;CHECK IF AN ADDRESS HERE
013242  001406                                  BEQ     3$
013244  027401  000000                          CMP     @(R4),R1                ;COMPARE DRIVE NUMBERS
013250  001412                                  BEQ     4$                      ;BRANCH IF A MATCH
013252  005724                          2$:     TST     (R4)+                   ;BUMP ADDRESS
013254  005302                                  DEC     R2
013256  001370                                  BNE     1$                      ;LOOK AT ALL OF THEM
013260                                  3$:     ERRDF 35,,ERR035                ;UNIT NUMBER NOT FOUND
013260  104455                                  TRAP    C$ERDF
013262  000043                                  .WORD   35
013264  000000                                  .WORD   0
013266  006710                                  .WORD   ERR035
013270                                          POP     R2
013272  000244                                  CLZ                             ;CLEAR Z AS ERROR FLAG
013274  000207                                  RETURN

013276  011404                          4$:     MOV     (R4),R4                 ;GET ADDRESS OF TABLE
013300  116437  000002  002074                  MOVB    D.UNIT(R4),L$LUN        ;GET UNIT NUMBER
013306                                          POP     R2
013310  000264                                  SEZ                             ;SET Z FLAG
013312  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

013314                          GETCNT: PUSH    R0
013316  005001                          CLR     R1                      ;START WITH ZERO COUNT
013320  121227  000060          GETCNX: CMPB    (R2),#'0                ;CHECK IF CHARACTER A DIGIT
013324  103415                          BLO     GETCDN                  ;BRANCH IF LOWER THAN ZERO
013326  121227  000071                  CMPB    (R2),#'9
013332  101012                          BHI     GETCDN                  ;BRANCH IF HIGHER THAN NINE
013334  006301                          ASL     R1                      ;MULTIPLY NUMBER BY 10
013336  010100                          MOV     R1,R0                   ; SAVE 2N
013340  006301                          ASL     R1                      ; COMPUTE 4N
013342  006301                          ASL     R1                      ; COMPUTE 8N
013344  060001                          ADD     R0,R1                   ; 8N + 2N = 10N
013346  112200                          MOVB    (R2)+,R0                ;GET DIGIT FROM STING
013350  162700  000060                  SUB     #'0,R0                  ;GET RID OF ASCII
013354  060001                          ADD     R0,R1                   ;ADD TO NUMBER
013356  000760                          BR      GETCNX                  ;GO TO NEXT CHARACTER
013360  005701                  GETCDN: TST     R1                      ;CHECK IF NUMBER IS ZERO
013362  001001                          BNE     GETCXX                  ;IF ZERO, CHANGE
013364  005201                          INC     R1                      ; TO DEFAULT OF ONE
013366                          GETCXX: POP     R0
013370  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 (LEFT JUSTIFIED).
                                        ;       R0 - CONTENTS DESTROYED

013372  010100                  PNTNUM: MOV     R1,R0           ; SAVE RADIX
013374  004737  013314                  CALL    GETCNT          ; GET COUNT OF BITS
013400                          PNTNUS: PUSH    <R2,R3,R5>      ; SAVE REGISTERS ON STACK
013406  012403                          MOV     (R4)+,R3        ; GET ONE PARAMETER WORD
013410  005005                          CLR     R5              ; CLEAR STORAGE FOR OTHER
013412  020127  000020                  CMP     R1,#16.         ; MORE THAN 16 BITS IN NUMBER?
013416  003401                          BLE     1$              ; NO, SKIP
013420  012405                          MOV     (R4)+,R5        ; YES, GET SECOND PARAMETER WORD
013422                          1$:     PUSH    R4              ; SAVE R4 ON STACK
013424  010504                          MOV     R5,R4           ; PUT LOW WORD IN R4
013426  012702  000020                  MOV     #16.,R2         ; COMPUTE BITS NOT WANTED
013432  160102                          SUB     R1,R2           ; BY SUBTRACTING BITS TO USE
013434  002002                          BGE     2$              ; FROM 16.
013436  062702  000020                  ADD     #16.,R2         ; IF NEGATIVE, ADD 16 FOR FIRST WORD
013442  001414                  2$:     BEQ     6$              ; IF ZERO, NO BITS NEED BE CLEARED
013444  012705  100000                  MOV     #BIT15,R5       ; START MASK WITH SIGN BIT SET
013450  005302                  3$:     DEC     R2              ; COUNT BITS IN MASK
013452  001402                          BEQ     4$              ;
013454  006205                          ASR     R5              ; SHIFT MORE BITS TO RIGHT
013456  000774                          BR      3$              ;
013460  020127  000020          4$:     CMP     R1,#16.         ; MORE THAN 16 BITS IN NUMBER?
013464  003402                          BLE     5$              ;
013466  040504                          BIC     R5,R4           ; YES, CLEAR IN HIGH WORD
013470  000401                          BR      6$              ;
013472  040503                  5$:     BIC     R5,R3           ; NO, CLEAR IN LOW WORD
013474  004737  013634          6$:     CALL    DIVIDE          ; DIVIDE BY RADIX IN R0
013500                                  PUSH    R5              ; PUSH REMAINDER ON STACK
013502  005202                          INC     R2              ; COUNT DIGITS ON STACK
013504  005703                          TST     R3              ; CHECK IF QUOTIENT IS ZERO
013506  001372                          BNE     6$              ;
013510  005704                          TST     R4              ;
013512  001370                          BNE     6$              ;
013514  020027  000012                  CMP     R0,#10.         ; IF RADIX IS DECIMAL
013520  001423                          BEQ     10$             ;       JUST GO PRINT DIGITS ON STACK
013522  010103                          MOV     R1,R3           ; OTHERWISE COMPUTE NUMBER OF LEADING 0
013524  162700  000014                  SUB     #12.,R0         ; DIVIDEND IS BITS IN NUMBER
013530  003002                          BGT     7$              ; DIVISOR IS BITS PER DIGIT PRINTED
013532  012700  000003                  MOV     #3,R0           ;       (3 OR 4)
013536  004737  013634          7$:     CALL    DIVIDE          ;
013542  005705                          TST     R5              ; IF REMAINDER NOT ZERO
013544  001401                          BEQ     8$              ; INCREMENT QUOTIENT
013546  005203                          INC     R3              ;
013550  160203                  8$:     SUB     R2,R3           ; SUBTRACT DIGITS ON STACK
```

```
 58 013552  001406                    BEQ     10$                     ; NO LEADING ZEROS IF ZERO
 59 013554                      9$:   PRINT   #'0                      ;
 60 013560  005303                    DEC     R3                      ;
 61 013562  001372                    BNE     9$                      ; REPEAT UNTIL COUNT REACHES ZERO
 62 013564                      10$:  POP     R5                      ; GET CHACACTER FROM STACK
 63 013566  062705  000060            ADD     #'0,R5                  ; CNVERT TO ASCII DIGIT
 64 013572  020527  000071            CMP     R5,#'9                  ; IF GREATER THAN A 9
 65 013576  003402                    BLE     11$                     ;       CONVERT TO A OR HIGHER
 66 013600  062705  000007            ADD     #<'A-'9-1>,R5           ;       FOR HEX DIGIT
 67 013604                      11$:  PRINT   R5                      ;
 68 013610  005302                    DEC     R2                      ; REPEAT FOR ALL DIGITS
 69 013612  001363                    BNE     10$                     ;       ON STACK
 70 013614                            POP     <R4,R5,R3,R2>           ;
 71 013624  000207                    RETURN                          ;
 72
```

```
                                ;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
                                
013634                          DIVIDE: PUSH    R2
013636  012702  000040                  MOV     #32.,R2                 ;SET UP SHIFT COUNT
013642  005005                          CLR     R5                      ;START WITH ZERO REMAINDER
013644  006303                  1$:     ASL     R3                      ;SHIFT LEFT INTO R5
013646  006104                          ROL     R4
013650  006105                          ROL     R5
013652  020005                          CMP     R0,R5                   ;WILL DIVISOR GO INTO REMAINDER
013654  101002                          BHI     2$                      ;ONLY SUBTRACT IF IT WILL
013656  160005                          SUB     R0,R5                   ;SUBTRACT DIVISOR
013660  005203                          INC     R3                      ;PUT A ONE INTO QUOTIENT
013662  005302                  2$:     DEC     R2                      ;COUNT THE SHIFTS
013664  001367                          BNE     1$
013666                                  POP     R2
013670  000207                          RETURN
```

```
 1
 2                                      ;DIV10
 3                                      ;
 4                                      ;DIVIDE A 64 BIT UNSIGNED NUMBER BY A 10.
 5                                      ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
 6                                      ;WILL NOT CHECK FOR DIVIDE BY ZERO.
 7                                      ;
 8                                      ;INPUTS:
 9                                      ;       R1 - LOW  16 BITS OF DIVIDEND
10                                      ;       R2 - NEXT 16 BITS OF DIVIDEND
11                                      ;       R3 - NEXT 16 BITS OF DIVIDEND
12                                      ;       R4 - HIGH 16 BITS OF DIVIDEND
13                                      ;OUTPUTS:
14                                      ;       R1 - QUOTIENT,
15                                      ;       R2 - QUOTIENT,
16                                      ;       R3 - QUOTIENT,
17                                      ;       R4 - QUOTIENT,
18                                      ;       R5 - REMAINDER
19
20 013672                              DIV10:  PUSH    R0                      ;DIVIDEND IS IN <R4,R3,R2,R1>
21 013674  012700  000100                      MOV     #64.,R0                 ;SET UP SHIFT COUNT
22 013700  005005                              CLR     R5                      ;START WITH ZERO REMAINDER
23 013702  006301                      1$:     ASL     R1
24 013704  006102                              ROL     R2                      ;SHIFT LEFT INTO R5
25 013706  006103                              ROL     R3
26 013710  006104                              ROL     R4
27 013712  006105                              ROL     R5
28 013714  022705  000012                      CMP     #10.,R5                 ;SILL DIVISOR GO INTO REMAINDER?
29 013720  101003                              BHI     2$                      ;ONLY SUBTRACT IF IT WILL
30 013722  162705  000012                      SUB     #10.,R5                 ;SUBTRACT DIVISOR
31 013726  005201                              INC     R1                      ;PUT A ONE INTO QUOTIENT
32 013730  005300                      2$:     DEC     R0                      ;COUNT THE SHIFTS
33 013732  001363                              BNE     1$
34 013734                                      POP     R0                      ;RETURN WITH QUOTIENT IN
35 013736  000207                              RETURN                          ;  <R4,R3,R2,R1> AND REMAINDER IN R5
36
```

PRE-PROGRAMMED SUBROUTINES

```
                                        ;BUILD DEFAULT 28-BIT NUMBER
                                        ;
                                        ;INPUT:
                                        ;       R4 - POINTER TO 2 WORD DEFAULT NUMBER
                                        ;OUTPUT:
                                        ;       TEMP - ASCIZ STRING REPRESENTING DEFAULT NUMBER
                                        ;
013740                                  BLD28:  PUSH <R0,R1,R3,R4,R5>
013752  011403                                  MOV (R4),R3                     ;GET NUMBER
013754  016404  000002                          MOV 2(R4),R4
013760  012700  000012                          MOV #10.,R0                     ;DIVISOR IS 10.
013764  005001                                  CLR R1                          ;CLEAR CHARACTER COUNT
013766  004737  013634                  1$:     CALL DIVIDE
013772  062705  000060                          ADD #'0,R5                      ;CONVERT REMAINDER TO ASCII CHARACTER
013776                                          PUSH R5                         ;STORE ON STACK
014000  005201                                  INC R1                          ;COUNT THE CHARACTER
014002  010305                                  MOV R3,R5                       ;REPEAT UNTIL QUOTIENT IS ZERO
014004  050405                                  BIS R4,R5
014006  001367                                  BNE 1$
014010  012700  002214                          MOV #TEMP,R0                    ;GET POINTER TO STRING
014014                                  2$:     POP R5                          ;PUT CHARACTERS INTO STRING
014016  110520                                  MOVB R5,(R0)+
014020  005301                                  DEC R1
014022  001374                                  BNE 2$
014024  105020                                  CLRB (R0)+                      ;END WITH NULL
014026                                          POP <R5,R4,R3,R1,R0>
014040  000207                                  RETURN
```

```
                                        ;CONVERT ASCIZ STRING TO 28-BIT NUMBER
                                        ;
                                        ;INPUTS:
                                        ;       TEMP - ASCIZ STRING UP TO 9 CHARACTERS LONK
                                        ;       R4 - ADDRESS OF TWO WORD STORAGE
                                        ;OUTPUTS:
                                        ;       IF STRING IS VALID NUMBER
                                        ;         TWO WORDS AT R4 LOADED WITH NUMBER
                                        ;         R4 POINTING TO WORD AFTER STORAGE
                                        ;         CARRY CLEAR
                                        ;       IF STRING INVALID
                                        ;         ERROR MESSAGE PRINTED
                                        ;         CARRY SET
                                        
014042                                  CNV28:  PUSH <R0,R1,R2,R3>
014052  005000                                  CLR R0                          ;START WITH ZEROS
014054  005001                                  CLR R1
014056  012702  002214                          MOV #TEMP,R2                    ;GET ADDRESS OF STRING
014062  112203                          1$:     MOVB (R2)+,R3                   ;GET A DIGIT FROM STRING
014064  001452                                  BEQ 3$                          ;IF NULL CHARACTER, ALL DONE
014066  162703  000060                          SUB #'0,R3                      ;SUBTRACT CHARACTER 0
014072  100435                                  BMI 2$
014074  022703  000011                          CMP #9.,R3
014100  103432                                  BLO 2$
014102  006300                                  ASL R0                          ;MULTIPLY BY 2
014104  006101                                  ROL R1
014106                                          PUSH <R1,R0>                    ;SAVE N X 2
014112  006300                                  ASL R0                          ;TIMES 2 AGAIN FOR N X 4
014114  006101                                  ROL R1
014116  006300                                  ASL R0                          ;TIMES 2 AGAIN FOR N X 8
014120  006101                                  ROL R1
014122  062600                                  ADD (SP)+,R0                    ;ADD N X 2 TO GIVE N X 10
014124  005501                                  ADC R1
014126  062601                                  ADD (SP)+,R1
014130  060300                                  ADD R3,R0                       ;ADD CURRENT DIGIT
014132  005501                                  ADC R1
014134  032701  170000                          BIT #170000,R1                  ;CHECK SIZE OF NUMBER
014140  001750                                  BEQ 1$                          ;MUST NOT BE MORE THAN 28 BITS
014142                                          PRINTF #INP28A                  ;PRINT PROPER RANGE
014142  012746  025557                          MOV     #INP28A,-(SP)
014146  012746  000001                          MOV     #1,-(SP)
014152  010600                                  MOV     SP,R0
014154  104417                                  TRAP    C$PNTF
014156  062706  000004                          ADD     #4,SP
014162  000261                                  SEC                             ;SET CARRY TO ASK AGAIN
014164  000415                                  BR 4$
                                        
014166                                  2$:     PRINTF  #INP28B                 ;PRINT ILLEGAL CHARACTER
014166  012746  025621                          MOV     #INP28B,-(SP)
014172  012746  000001                          MOV     #1,-(SP)
014176  010600                                  MOV     SP,R0
014200  104417                                  TRAP    C$PNTF
014202  062706  000004                          ADD     #4,SP
014206  000261                                  SEC
014210  000403                                  BR 4$
                                        
014212  010024                          3$:     MOV R0,(R4)+                    ;MOVE NUMBER TO STORAGE AREA
```

PRE-PROGRAMMED SUBROUTINES

```
     48 014214  010124                      MOV R1,(R4)+
     49 014216  000241                      CLC                             ;CLEAR CARRY TO INDICATE ALL IS WELL
     50 014220                        4$:   POP <R3,R2,R1,R0>
     51 014230  000207                      RETURN
    150
```

```
                                        ;++
                                        ;       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
                                        ;--

014232                          PNTERR: PUSH    <R0,R1,R2>              ; SAVE REGISTERS ON STACK
014240  005764  000004                  TST     4(R4)                   ; CHECK IF BIT 15 SET
014244  002004                          BGE     1$                      ; IF SO, GET UNIT FROM MESSAGE DATA
014246  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; ELSE, GET UNIT FROM CONTROLLER TABLE
014254  000417                          BR      2$                      ;
014256                          1$:     PUSH    R4                      ; SAVE DATA ADDRESS
014260  016401  000004                  MOV     4(R4),R1                ; GET DRIVE NUMBER
014264  004737  013224                  CALL    GTDRVT                  ; GET DRIVE TABLE ADDRESS
014270  001037                          BNE     5$                      ; IF UNIT DROPPED, EXIT
014272  005764  000002                  TST     D.UNIT(R4)              ; IS TESTING DONE ON UNIT?
014276  100005                          BPL     3$                      ; IF NOT DONE, PROCEED
014300  052713  100000                  BIS     #BIT15,(R3)             ; MARK UNIT AS DONE TESTING
014304                                  DORPT                           ; PRINT A STATISTICAL REPORT
014304  104424                          TRAP    C$DRPT
014306                                  POP     R4                      ; RESTORE DATA ADDRESS
014310  000423                          BR      4$                      ;

014312                          3$:     POP     R4                      ; RESTORE DATA ADDRESS
014314  012702  002146          2$:     MOV     #ERRTYP,R2              ; GET POINTER TO ERROR TABLE
014320  016412  000002                  MOV     2(R4),(R2)              ; GET ERROR TYPE
014324  006112                          ROL     (R2)                    ;
014326  006112                          ROL     (R2)                    ;
014330  006112                          ROL     (R2)                    ;
014332  042722  177774                  BIC     #^C3,(R2)+              ; CLEAR LOW 2 BITS
014336  016412  000002                  MOV     2(R4),(R2)              ;
014342  042722  140000                  BIC     #140000,(R2)+           ; MASK LOW 14 BITS
014346  005022                          CLR     (R2)+                   ; CLEAR MESSAGE POINTER
014350  012712  007160                  MOV     #ERR.TN,(R2)            ; GET ROUTINE NUMBER
014354                                  ERROR                           ; PRINT THE ERROR MESSAGE
014354  104460                          TRAP    C$ERROR
014356  000241                          CLC                             ; DRIVE HAS NOT BEEN DROPPED
014360                          4$:     POP     <R2,R1,R0>              ; RESTORE REGISTERS
014366  000207                          RETURN                          ;
014370                          5$:     POP     <R4>                    ; RESTORE STACK
014372  000261                          SEC                             ; DRIVE HAS BEEN DROPPED
014374  000771                          BR      4$                      ;
```

```
                                ;++
                                ;       LOADDM - LOAD AND START A DM PROGRAM IN 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
                                ;--

014376                          LOADDM:
014376  016504  000004                  MOV     C.VEC(R5),R4            ; GET VECTOR OF CONTROLLER
014402  042704  177000                  BIC     #^C<CT.VEC>,R4          ; CLEAR ALL BUT VECTOR
014406  010501                          MOV     R5,R1                   ; GET INTERRUPT SERVICE LINK
014410  062701  000006                  ADD     #C.JSR,R1               ;
014414                                  SETVEC  R4,R1,#PRIO7            ; SET UP INTERRUPT VECTOR
014414  012746  000340                  MOV     #PRIO7,-(SP)
014420  010146                          MOV     R1,-(SP)
014422  010446                          MOV     R4,-(SP)
014424  012746  000003                  MOV     #3,-(SP)
014430  104437                          TRAP    C$SVEC
014432  062706  000010                  ADD     #10,SP
014436  006204                          ASR     R4                      ; INITIALIZE CONTROLLER WITH SMALLEST
014440  006204                          ASR     R4                      ;       POSITION VECTOR FOR RING
014442  004737  015272                  CALL    CNTINT                  ;       BUFFER AND INTERRUPTS ENABLED
014446  001447                          BEQ     LOADER                  ; IF ERROR, EXIT
014450  004737  007356                  CALL    HCOMM                   ; ALLOCATE SPACE FOR HOST COMM AREA

014454  017701  165516          LOADTX: MOV     @DMPROG,R1              ; GET SIZE OF PROGRAM
014460  012700  000002          LOADB:  MOV     #OP.ESP,R0              ; BUILD EXECUTE SUPPLIED
014464  004737  014572                  CALL    BLDCMD                  ;       PROGRAM COMMAND PACKET
014470  013764  002176  000040          MOV     DMPROG,MC.CPK+P.UADR(R4); LOAD MAIN PROGRAM ADDRESS
014476  010164  000034                  MOV     R1,MC.CPK+P.BCNT(R4)    ;       AND SIZE
014502  013764  002176  000054          MOV     DMPROG,MC.CPK+P.OVRL(R4); LOAD OVERLAY ADDRESS
014510  067764  165462  000054          ADD     @DMPROG,MC.CPK+P.OVRL(R4)
014516  004737  014664                  CALL    SNDCMD                  ; SEND COMMAND TO CONTROLLER
014522  004737  014774                  CALL    WAITMS                  ; WAIT FOR MESSAGE RESPONSE
014526  001417                          BEQ     LOADER                  ; EXIT IF ERROR
014530  032764  000037  000032          BIT     #ST.MSK,MC.MPK+P.STS(R4); CHECK FOR ERRORS
014536  001007                          BNE     LOADE1                  ;
014540  042765  000024  000012          BIC     #CT.CMD+CT.REQ,C.FLG(R5); CLEAR COMMAND OUTSTANDING FLAG
014546  052765  000002  000012          BIS     #CT.RN,C.FLG(R5)        ; SET DM PROGRAM RUNNING FLAG

014554  000207                          RETURN                          ; SUCCESS RETURN
                                ;
                                ;       CONTROLLER FAILED TO DOWNLINE LOAD DM PROGRAM
                                ;

014556                          LOADE1: ERRDF   33,,ERR033              ;
014556  104455                          TRAP    C$ERDF
014560  000041                          .WORD   33
014562  000000                          .WORD   0
014564  006702                          .WORD   ERR033
```

PRE-PROGRAMMED SUBROUTINES

```
     92 014566  000264                LOADER: SEZ                     ; SET Z TO INDICATE ERROR OCCURRED
     93 014570  000207                        RETURN                  ; ERROR RETURN
     94
```

```
 1
 2                                      ;**
 3                                      ;       BLDCMD - BUILD A COMMAND IN COMMAND PACKET
 4                                      ;
 5                                      ;       INPUTS:
 6                                      ;               R5 - CONTROLLER TABLE ADDRESS
 7                                      ;               R0 - COMMAND CODE
 8                                      ;       OUTPUTS:
 9                                      ;               R4 - ADDRESS OF HOST COMM AREA
10                                      ;               COMMAND PACKET CONTAINING REF NUMBER AND OPCODE. ALL
11                                      ;                       OTHER FIELDS CLEARED.
12                                      ;               COMMAND REFERRENCE NUMBER IN CONTROLLER TABLE INCREMENTED
13                                      ;                       AND RESULT IN COMMAND PACKET.
14                                      ;               R0 - CONTENTS DESTROYED
15                                      ;--
16
17 014572                               BLDCMD: PUSH    <R1,R0>                 ; SAVE REGISTERS ON STACK
18 014576  016504  000014                       MOV     C.HCOM(R5),R4           ; GET ADDRESS OF HOST COMM AREA
19 014602  010400                               MOV     R4,R0                   ; COPY TO R0
20 014604  062700  000014                       ADD     #HC.CEV,R0              ; COMPUTE ADDRESS OF COMMAND ENVELOPE
21 014610  012720  000060                       MOV     #HC.PSZ,(R0)+           ; LOAD PACKET LENGTH
22 014614  012701  001000                       MOV     #DUP,R1                 ; LOAD DUP CIRCUIT IDENTIFIER
23 014620  022716  000031                       CMP     #OP.MWR,(SP)            ; IF CODE IS MAINTENANCE WRITE
24 014624  001403                               BEQ     1$                      ;       USE DIAGNOSTIC CIRCUIT ID
25 014626  022716  000030                       CMP     #OP.MRD,(SP)            ; IF CODE IS NOT MAINTENANCE READ
26 014632  001002                               BNE     BLDC0                   ;       SKIP
27 014634  012701  177777           1$:         MOV     #DIAG,R1                ; ELSE, USE DIAGNOSTIC CIRCUIT ID
28 014640  010120                   BLDC0:      MOV     R1,(R0)+                ; PUT IDENTIFIER INTO PACKET
29 014642  012701  000030                       MOV     #<HC.PSZ>/2,R1          ; GET WORDS TO CLEAR
30 014646  005020                   BLDC1:      CLR     (R0)+                   ; CLEAR PACKET
31 014650  005301                               DEC     R1                      ; ANY MORE
32 014652  001375                               BNE     BLDC1                   ;       WORDS TO CLEAR?
33 014654                                       POP     HC.CPK+P.OPCD(R4)       ; PUT OPCODE IN PACKET
34 014660                                       POP     R1                      ; RESTORE R1
35 014662  000207                               RETURN
36
```

```
                                ;++
                                ;       SNDCMD - SEND A COMMAND TO THE CONTROLLER. CLEAR THE RESPONSE
                                ;       PACKET. MARK BOTH PACKETS AVAILABLE TO THE CONTROLLER. SET COMMAND
                                ;       ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE TIMEOUT COUNTER.
                                ;
                                ;       INPUTS:
                                ;               R5 - CONTROLLER TABLE ADDRESS
                                ;       OUTPUTS:
                                ;               R4 - ADDRESS OF HOST COMM AREA
                                ;--

014664  016504  000014          SNDCMD: MOV     C.HCOM(R5),R4           ; LOAD R4 WITH HOST COMM AREA ADDRESS
014670  005265  000032                  INC     C.REF(R5)               ; INCREMENT CMD REFERENCE NUMBER
014674  016564  000032  000020          MOV     C.REF(R5),HC.CPK+P.CRF(R4); PUT IN PACKET
014702  012764  140000  000006          MOV     #RG.OWN+RG.FLG,HC.MCT(R4) ; MARK MESSAGE PACKET AVAILABLE
014710  012764  100000  000012          MOV     #RG.OWN,HC.CCT(R4)      ; MARK COMMAND TO CONTROLLER
014716  005775  000000                  TST     @(R5)                   ; TELL CONTROLLER COMMAND IS THERE
014722  052765  000004  000012          BIS     #CT.CMD,C.FLG(R5)       ; MARK COMMAND ISSUED
014730  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
                                        ;--

014732                          CLRBUF: PUSH    <R0,R1>                 ; SAVE REGISTERS ON STACK
014736  060400                          ADD     R4,R0                   ; CREATE BUFFER ADDRESS
014740  010064  000040                  MOV     R0,HC.CPK+P.UADR(R4)    ; PUT BUFFER ADDRESS IN COMMAND PACKET
014744  012764  000106  000034          MOV     #HC.BSZ,HC.CPK+P.BCNT(R4);PUT SIZE OF BUFFER IN COMMAND PACKET
014752  010004                          MOV     R0,R4                   ; PUT BUFFER ADDRESS IN R4
014754  012701  000043                  MOV     #<HC.BSZ>/2,R1          ; GET SIZE OF BUFFER IN WORDS
014760  005020                  CLRBFL: CLR     (R0)+                   ; CLEAR ALL THE WORDS
014762  005301                          DEC     R1                      ; ANY MORE
014764  001375                          BNE     CLRBFL                  ;       WORDS TO CLEAR?
014766                                  POP     <R1,R0>                 ;
014772  000207                          RETURN                          ;
```

```
                                        ;++
                                        ;       WAITMS - WAIT FOR CONTROLLER TO RESPOND WITH A MESSAGE PACKET
                                        ;
                                        ;       INPUTS:
                                        ;               R5 - ADDRESS OF CONTROLLER TABLE
                                        ;       OUTPUTS:
                                        ;               Z CLEAR IF NO ERROR
                                        ;               Z SET IF ERROR, MESSAGE PRINTED
                                        ;--

014774                          WAITMS: PUSH    <R0,R1>                 ; SAVE REGISTERS ON STACK
015000  012700  000036                  MOV     #30.,R0                 ; SET TIME OUT VALUE OF 30 SECONDS
015004  010501                          MOV     R5,R1                   ; POINT TO TIME OUT COUNTER
015006  062701  000026                  ADD     #C.TO,R1                ; POINTER TO TIMER FIELD
015012  004737  015210                  CALL    SETTO                   ; START TIMER
015016  011500                  1$:     MOV     (R5),R0                 ; GET ADDRESS OF IP REGISTER
015020  032765  000010  000012          BIT     #CT.MSG,C.FLG(R5)       ; LOOK IF INTERRUPT OCCURRED
015026  001025                          BNE     3$                      ; BRANCH IF SO
015030  016001  000002                  MOV     2(R0),R1                ; LOOK AT SA REGISTER
015034  001031                          BNE     4$                      ; BRANCH IF ERROR CODE PRESENT
27 015036                               BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
   015036  104422                       TRAP    C$BRK
28 015040  005737  002320               TST     KW.CSR                  ; SEE IF A CLOCK ON SYSTEM
29 015044  001764                       BEQ     1$                      ; IF NOT, DON'T TIMEOUT
30 015046  023765  002332  000030       CMP     KW.EL+2,C.TOH(R5)       ; CHECK HIGH WORD OF INTERVAL
31 015054  101005                       BHI     2$                      ; IF GREATER, TIMED OUT
32 015056  001357                       BNE     1$                      ; IF NOT EQUAL, NO TIMEOUT OCCURRED
33 015060  023765  002330  000026       CMP     KW.EL,C.TO(R5)          ; CHECK LOW WORD OF INTERVAL
34 015066  103753                       BLO     1$                      ; IF LOWER, NO TIMEOUT OCCURRED
35 015070                       2$:     ERRDF   36.,ERR036              ; PRINT TIMEOUT ERROR
   015070  104455                       TRAP    C$ERDF
   015072  000044                       .WORD   36
   015074  000000                       .WORD   0
   015076  006726                       .WORD   ERR036
36 015100  000416                       BR      5$                      ; ERROR EXIT
37
38 015102  042765  000010  000012  3$:     BIC     #CT.MSG,C.FLG(R5)       ; CLEAR MESSAGE RECEIVED FLAG
39 015110                               POP     <R1,R0>                 ; SAVE REGISTERS ON STACK
40 015114  000244                       CLZ                             ; GIVE NO ERROR RETURN
41 015116  000207                       RETURN                          ;
42 015120                       4$:
46 015120                               PUSH    R2                      ; SAVE R2
47 015122  010102                       MOV     R1,R2                   ; GET SA REGISTER CONTENTS
48 015124                               ERRDF   40.,ERR040              ; CONTROLLER DETECTED ERROR
   015124  104455                       TRAP    C$ERDF
   015126  000050                       .WORD   40
   015130  000000                       .WORD   0
   015132  006752                       .WORD   ERR040
49 015134                               POP     R2                      ; RESTORE REGISTERS
51 015136                       5$:     POP     <R1,R0>                 ; RESTORE REGISTERS
52 015142  000264                       SEZ                             ; Z SET OR ERROR RETURN
53 015144  000207                       RETURN                          ;
```

```
 1
 2                                      ;++
 3                                      ;       NXMI - NON-EXISTANT MEMORY SERVICE ROUTINE
 4                                      ;
 5                                      ;       INPUTS:
 6                                      ;               NXMAD SET TO ZERO
 7                                      ;       OUTPUTS:
 8                                      ;               NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
 9                                      ;--
10
11 015146                              BGNSRV  NXMI
   015146                              NXMI::
12 015146  012737  177777  002340              MOV     #-1,NXMAD
13 015154                              ENDSRV
   015154                              L10020:
   015154  000002                              RTI
14
15                                      ;++
16                                      ;       CNTSRV - CONTROLLER INTERRUPT SERVICE ROUTINE. MARKS CONTROLLER TABLE
17                                      ;       THAT AN INTERRUPT HAS BEEN RECEIVED.
18                                      ;
19                                      ;       THIS ROUTINE IS CALLED BY A [JSR R0,CNTSRV] INSTRUCTION FROM WITHIN
20                                      ;       THE CONTROLLER TABLE. THE PC STORED IN R0 IS THE ADDRESS OF THE C.FLG
21                                      ;       WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
22                                      ;       OF R0 FOLLOWED BY THE INTERRUPTED PC AND PS.
23                                      ;
24                                      ;       INPUTS:
25                                      ;               R0 - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
26                                      ;               STACK - SAVED CONTENTS OF R0
27                                      ;       OUTPUTS:
28                                      ;               CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
29                                      ;               R0 - RESTORED FROM STACK
30                                      ;--
31
32 015156                              BGNSRV  CNTSRV
   015156                              CNTSRV::
33 015156  052710  000010                      BIS     #CT.MSG,(R0)            ; SET CT.MSG
34 015162                                      POP     R0                      ; RESTORE R0
35 015164                              ENDSRV
   015164                              L10021:
   015164  000002                              RTI
36
46
47                                      ;++
48                                      ;       KW11I - CLOCK INTERRUPT SERVICE ROUTINE
49                                      ;--
50 015166                              BGNSRV  KW11I
   015166                              KW11I::
51 015166  062737  000001  002330              ADD     #1,KW.EL                ; COUNT THE INTERRUPT
52 015174  005537  002332                      ADC     KW.EL+2                 ; PUT CARRY IN HIGH WORD
53 015200  012777  000105  165112              MOV     #KW.OUT,@KW.CSR         ; RESTART THE CLOCK
54 015206                              ENDSRV
   015206                              L10022:
   015206  000002                              RTI
55
```

```
                                ;++
                                ;       SETTO - SET TIMEOUT COUNTER TO A GIVEN 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
                                ;--

015210                          SETTO:  PUSH    <R2,R3>
015214  005002                          CLR     R2                      ; CLEAR PRODUCT
015216  013703  002326                  MOV     KW.HZ,R3                ; GET MULTIPLICAND

015222  006200                  SETTOO: ASR     R0                      ; SHIFT MULTIPLIER TO RIGHT
015224  103001                          BCC     SETTO1                  ; IF A ONE BIT SHIFTED OUT
015226  060302                          ADD     R3,R2                   ;   ADD MULTIPLICAND TO PRODUCT
015230  006303                  SETTO1: ASL     R3                      ; DOUBLE THE MULTIPLICAND
015232  005700                          TST     R0                      ;
015234  001372                          BNE     SETTOO                  ; CONTINUE UNTIL MULTIPLIER IS ZERO

                                ;
                                ;       GET CURRENT TIME
                                ;

015236  013700  002330          SETT02: MOV     KW.EL,R0                ; GET TIME (LOW WORD)
015242  013703  002332                  MOV     KW.EL+2,R3              ; GET TIME (HIGH WORD)
015246  020037  002330                  CMP     R0,KW.EL                ; IF CHANGED DURING RETRIEVAL
015252  001371                          BNE     SETT02                  ;   GET IT AGAIN

                                ;
                                ;       ADD TIME TIL TIMEOUT
                                ;

015254  060200                          ADD     R2,R0                   ; ADD TIMEOUT TO CURRENT TIME
015256  005503                          ADC     R3                      ; INCREMENT HIGH WORD IF CARRY

                                ;
                                ;       PUT RESULT IN STORAGE
                                ;

015260  010021                          MOV     R0,(R1)+                ; SAVE LOW WORD OF TIMEOUT
015262  010311                          MOV     R3,(R1)                 ; SAVE HIGH WORD OF TIMEOUT

015264                                  POP     <R3,R2>
015270  000207                          RETURN
```

```
                                ;**
                                ;       CNTINT
                                ;
                                ;       FUNCTIONAL DESCRIPTION:
                                ;               SUBROUTINE TO INITIALIZE A CONTROLLER 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 - LENGTH, INTERRUPT AND VECTOR FIELDS TO SEND TO CONTROLLER
                                ;
                                ;       IMPLICIT INPUTS:
                                ;               FFREE - FIRST FREE ADDRESS OF MEMORY. THIS ADDRESS IS GIVEN TO
                                ;                       CONTROLLER AS START OF RING BUFFER.
                                ;               FSIZE - SIZE OF FREE MEMORY AVAILABLE IN WORDS.
                                ;
                                ;       OUTPUTS:
                                ;               R1 - SIZE OF RING BUFFER IN WORDS IF NO ERROR
                                ;               R4 - ADDRESS OF IP REGISTER IN CONTROLLER
                                ;               R5 - UNCHANGED
                                ;               Z - CLR IF NO ERROR, SET IF ANY ERROR REPORTED
                                ;--
                                ;
                                ;       CHECK IF ENOUGH FREE MEMORY FOR RING BUFFER
                                ;
015272                          CNTINT: PUSH    <R3>                    ; SAVE R3 ON STACK
015274  012701  000006                  MOV     #6.,R1                  ; GET SIZE OF RING BUFFER
015300  020137  002160                  CMP     R1,FSIZE                ; COMPARE WITH SIZE OF FREE MEMORY
015304  101402                          BLOS    1$                      ; THERE IS PLENTY OF MEMORY
015306  000137  007320                  JMP     FMERR                   ; FATAL ERROR IF NOT ENOUGH MEMORY
015312                          1$:
                                ;
                                ;       DO THE INITIALIZATION
                                ;
015312  004737  015502                  CALL    CNTIST                  ; DO FIRST THREE STEPS
015316  103466                          BCS     9$                      ; EXIT IF MICROCODE REPORTED FAILURE
015320  012364  000002                  MOV     (R3)+,2(R4)             ; WRITE STEP 3 RESPONSE TO SA REGISTER
015324  004737  015776                  CALL    CNTRSP                  ; WAIT FOR STEP OR ERROR BIT
015330  103461                          BCS     9$                      ; EXIT IF MICROCODE REPORTED FAILURE
015332  010237  002362                  MOV     R2,SSTEP4               ; SAVE STEP 4 RESPONSE VALUE.
                                ;
                                ;       SEND GO BIT TO SA REGISTER TO END INITIALIZATION
                                ;
015336                          8$:
015336  012700  000001                  MOV     #SA.GO,R0               ; SET THE GO BIT
015342  010064  000002                  MOV     R0,2(R4)                ; WRITE TO SA REGISTER
                                ;
                                ;       NOW CHECK IF THE CONTROLLER TYPE IS VALID AND MICROCODE VERSION
                                ;       IS CURRENT
                                ;
015346  013703  002362          23$:    MOV     SSTEP4,R3               ; GET SAVED CONTROLLER STEP 4 RESPONSE
015352  010301                          MOV     R3,R1                   ; R3 HAS STEP 4 INFO
015354  042701  177760                  BIC     #^C<SA.MCV>,R1          ; R1 = MICRO CODE LEVEL
```

```
                                        ROR     R3                      ; RIGHT
                                        ROR     R3                      ;      JUSTIFY
                                        ROR     R3                      ;            MODEL
                                        ROR     R3                      ;                  NUMBER
112 015360  006003                      BIC     #↑C<SA.CNT/16.>,R3      ; R3 = CONTROLLER MODEL NUMBER
113 015362  006003
114 015364  006003
115 015366  006003
116 015370  042703  177760              CMP     #6.,R3                  ; CHECK IF UDA50A
117 015374  022703  000006              BEQ     24$                     ; IF SO, BRANCH
118 015400  001413                      CMP     #13.,R3                 ; CHECK IF KDA50-Q
119 015402  022703  000015              BEQ     25$                     ; IF SO BRANCH
120 015406  001420                      BIS     #BIT15,C.UNIT(R5)       ;      ELSE MARK AS DO NOT EXECUTE
121 015410  052765  100000  000002      ERRDF   14.,ERR014              ;      REPORT ERROR
122 015416                              TRAP    C$ERDF
    015416  104455                      .WORD   14
    015420  000016                      .WORD   0
    015422  000000                      .WORD   ERR014
    015424  006620
123 015426  000422                      BR      9$                      ;      EXIT
124 015430                      24$:
126 015430  020127  000003              CMP     R1,#3.                  ; UDA50A MICROCODE VERSION UP TO DATE?
127 015434  002014                      BGE     26$                     ; IF SO, BRANCH
128 015436                              ERRSF   14.,ERR014              ;      ELSE, REPORT ERROR
    015436  104454                      TRAP    C$ERSF
    015440  000016                      .WORD   14
    015442  000000                      .WORD   0
    015444  006620                      .WORD   ERR014
129 015446  000407                      BR      26$                     ;
130 015450                      25$:
132 015450  020127  000000              CMP     R1,#0.                  ; KDA50-Q MICROCODE VERSION UP TO DATE?
133 015454  002004                      BGE     26$                     ; IF SO, BRANCH
134 015456                              ERRSF   14.,ERR014              ;      ELSE, REPORT ERROR
    015456  104454                      TRAP    C$ERSF
    015460  000016                      .WORD   14
    015462  000000                      .WORD   0
    015464  006620                      .WORD   ERR014
135 015466                      26$:    POP     <R3>                    ; RESTORE R3 FROM STACK
136 015470  000244                      CLZ                             ; CLEAR Z TO INDICATE NO ERROR
137 015472  000207                      RETURN                          ;
138
139                                     ;
140                                     ;       ERROR RETURN
141                                     ;
142
143 015474                      9$:     POP     <R3>                    ; RESTORE R3 FROM STACK
144 015476  000264                      SEZ                             ; SET Z TO INDICATE ERROR OCCURRED
145 015500  000207                      RETURN                          ;
146
```

```
                                        ;**
                                        ;       CNTIST
                                        ;
                                        ;       START THE INITIALIZATION PROCESS ON THE SELECTED CONTROLLER.
                                        ;       STOP BEFORE WRITING THE THIRD WORD SO CONTROLLER DOES NOT
                                        ;       ATTEMPT ANY BUS TRANSFERS.
                                        ;
                                        ;       INPUTS:
                                        ;               R5 - ADDRESS OF CONTROLLER TABLE
                                        ;               R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO CONTROLLER
                                        ;
                                        ;       LOAD TABLE OF DATA TO SEND TO SA REGISTER
                                        ;--

015502                                  CNTIST:
015502                                          BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
015502  104422                                  TRAP    C$BRK
015504                                          PUSH    R1                      ;
015506  052704  100000                          BIS     #SA.STP,R4              ; SET STEP BIT IN DATA WORD
015512  010437  002350                          MOV     R4,SND.S1               ; LOAD SEND DATA FOR STEP 1 OF INIT
015516  013737  002156  002354                  MOV     FFREE,SND.S2            ; GET MEMORY ADDRESS AND
015524  062737  000004  002354                  ADD     #MC.MSG,SND.S2          ; LOAD SEND DATA FOR STEP 2 OF INIT
015532  012737  000000  002360                  MOV     #SA.TST,SND.S3          ; LOAD SEND DATA FOR STEP 3 OF INIT
                                        ;
                                        ;       VERIFY THE ADDRESS OF THE SA AND IP REGISTERS ARE VALID AND
                                        ;       START THE INITIALIZATION BY WRITING ZEROS TO IP REGISTER
                                        ;
015540  016504  000000                          MOV     C.UADR(R5),R4           ; GET ADDRESS OF IP REGISTER
015544  005037  002340                          CLR     NXMAD                   ; CLEAR MEMORY ERROR FLAG
015550                                          SETVEC  #ERRVEC,#NXMI,#PRI07    ; SETUP TIMEOUT ERROR VECTOR
015550  012746  000340                          MOV     #PRI07,-(SP)
015554  012746  015146                          MOV     #NXMI,-(SP)
015560  012746  000004                          MOV     #ERRVEC,-(SP)
015564  012746  000003                          MOV     #3,-(SP)
015570  104437                                  TRAP    C$SVEC
015572  062706  000010                          ADD     #10,SP
015576  005764  000002                          TST     2(R4)                   ; ACCESS SA REGISTER
015602  005014                                  CLR     (R4)                    ; WRITE TO IP
015604                                          CLRVEC  #ERRVEC                 ; RETURN TIMEOUT ERROR VECTOR
015604  012700  000004                          MOV     #ERRVEC,R0
015610  104436                                  TRAP    C$CVEC
015612  005737  002340                          TST     NXMAD                   ; SEE IF A MEMORY ERROR OCCURRED
015616  001406                                  BEQ     1$                      ; IF NO ERROR, CONTINUE
015620                                          ERRDF   38.,.ERR038             ; PRINT ERROR MESSAGE
015620  104455                                  TRAP    C$ERDF
015622  000046                                  .WORD   38
015624  000000                                  .WORD   0
015626  006740                                  .WORD   ERR038
015630  000261                                  SEC                             ; CARRY SET TO INDICATE ERROR
015632  000424                                  BR      4$                      ; EXIT

                                        ;
                                        ;       SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
                                        ;

015634  012737  004000  002364  1$:             MOV     #SA.S1,CNTRSD           ; STORE RESPONSE MASK
```

```
 45 015642  012703  002346                    MOV     #INITBL,R3              ; GET INDEX TO SEND/REPOND INIT TABLE
 46
 47                                  ;
 48                                  ;       WAIT FOR AND CHECK RESPONSE DATA
 49                                  ;
 50
 51 015646  004737  015776          2$:      CALL    CNTRSP                  ; WAIT FOR STEP OR ERROR BITS
 52 015652  103414                           BCS     4$                      ; EXIT IF ERROR
 53 015654  004733                           CALL    @(R3)+                  ; CALL RESPONSE CHECKER FOR STEP
 54 015656  103412                           BCS     4$                      ; GET OUT IF ERROR
 55 015660  006337  002364                    ASL     CNTRSD                  ; SHIFT TO NEXT STEP BIT
 56 015664  032737  040000  002364           BIT     #SA.S4,CNTRSD           ; CHECK IF NOW AT STEP 4
 57 015672  001003                           BNE     3$                      ; GET OUT IF SO
 58 015674  012364  000002                    MOV     (R3)+,2(R4)             ; WRITE DATA TO SA REGISTER
 59 015700  000762                            BR      2$                      ; STAY IN LOOP
 60
 61 015702  000241                  3$:       CLC                             ; CLEAR CARRY FOR NO ERROR INDICATION
 62 015704                          4$:       POP     R1                      ; RESTORE R1
 63 015706  000207                            RETURN                          ;
 64                                  ;
 65                                  ;       RESPONSE CHECK FOR FIRST WORD (STEP 1) FROM SA REGISTER
 66                                  ;       CHECK FOR PROPER CONTROLLER TYPE
 67                                  ;
 68 015710  012701  004400          RSP.S1:  MOV     #SA.S1+SA.DI,R1         ; SET STEP ONE BIT
 69 015714  042702  001140                   BIC     #SA.QB+SA.MP+SA.SM,R2   ; CLEAR Q22 & SM BIT FOR KDA50-Q
 70 015720  000416                           BR      RSP.CK                  ; NOW DO A RESPONSE CHECK
 71                                  ;
 72                                  ;       RESPONSE CHECK FOR SECOND WORD (STEP 2) FROM SA REGISTER
 73                                  ;       CHECK FOR ECHO OF INTERRUPT ENABLE FLAG AND INTERRUPT VECTOR
 74                                  ;
 75 015722  013701  002350          RSP.S2:  MOV     SND.S1,R1               ; GET WORD SENT TO SA REGISTER
 76 015726  000301                           SWAB    R1                      ; GET HIGH 8 BITS
 77 015730  042701  177400                   BIC     #177400,R1              ;
 78 015734  052701  010000                   BIS     #SA.S2,R1               ; SET STEP 2 BIT
 79 015740  000406                           BR      RSP.CK                  ; NOW DO A RESPONSE CHECK
 80                                  ;
 81                                  ;       RESPONSE CHECK FOR THIRD WORD (STEP 3) FROM SA REGISTER
 82                                  ;       CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
 83                                  ;
 84 015742  013701  002350          RSP.S3:  MOV     SND.S1,R1               ; GET WORD SENT TO SA REGISTER
 85 015746  042701  177400                   BIC     #177400,R1              ; JUST LOW 8 BITS
 86 015752  052701  020000                   BIS     #SA.S3,R1               ; SET STEP 3 BIT
 87                                  ;
 88                                  ;       RESPONSE CHECK, COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
 89                                  ;
 90 015756  020102                  RSP.CK:  CMP     R1,R2                   ; COMPARE THE DATA
 91 015760  001405                           BEQ     1$                      ; EXIT IF COMPARED CORRECTLY
 95 015762                                   ERRDF   40,,ERR040              ; ERROR - WRONG DATA IN SA REGISTER
    015762  104455                           TRAP    C$ERDF
    015764  000050                           .WORD   40
    015766  000000                           .WORD   0
    015770  006752                           .WORD   ERR040
 97 015772  000261                           SEC                             ; SET CARRY TO INDICATE ERROR
 98 015774  000207                  1$:      RETURN                          ;
 99
```

```
                                        ;++
                                        ;       CNTRSP
                                        ;
                                        ;       WAIT FOR CONTROLLER TO RESPOND WITH DATA IN SA REGISTER.
                                        ;       EITHER STEP BIT FROM MASK IN LOCATION CNTRSD OR ERROR BIT
                                        ;       WILL CAUSE A TERMINATION.
                                        ;       AN ERROR MESSAGE WILL BE PRINTED IF THE CONTROLLER DOES NOT RESPOND
                                        ;       IN 10 SECONDS OR IF ERROR SETS.
                                        ;
                                        ;       INPUTS:
                                        ;               CNTRSD - MASK OF STEP BIT TO LOOK FOR
                                        ;               R5 - ADDRESS OF CONTROLLER TABLE
                                        ;               R4 - ADDRESS OF IP REGISTER
                                        ;       OUTPUTS:
                                        ;               ERROR MESSAGE IF TIME OUT ON RESPONSE OR ERROR BIT SETS
                                        ;               R2 - DATA FROM SA REGISTER
                                        ;               CARRY SET IF ERROR BIT SETS OR TIME OUT
                                        ;--

015776                          CNTRSP: PUSH    R1
016000  052737  100000  002364          BIS     #SA.ERR,CNTRSD          ; SET ERROR BIT IN MASK WORD
016006  012700  000012                  MOV     #10.,R0                 ; SET UP FOR 10 SECOND TIMEOUT
016012  010501                          MOV     R5,R1                   ; POINT TO COUNTER IN CONTROLLER TABLE
016014  062701  000026                  ADD     #C.TO,R1                ;
016020  004737  015210                  CALL    SETTO                   ;
016024                                  POP     R1                      ;
016026  033764  002364  000002  1$:     BIT     CNTRSD,2(R4)            ; LOOK AT ERROR AND STEP BIT
016034  001024                          BNE     3$                      ; BRANCH IF EITHER SET
016036                                  BREAK                           ; >>>>>>>>>BREAK BACK TO MONITOR<<<<<<<<
016036  104422                          TRAP    C$BRK
016040  005737  002320                  TST     KW.CSR                  ; SEE IF CLOCK ON SYSTEM
016044  001770                          BEQ     1$                      ;
016046  023765  002332  000030          CMP     KW.EL+2,C.TOH(R5)       ; CHECK IF TIME OUT OCCURRED
016054  101005                          BHI     2$                      ;
016056  001363                          BNE     1$                      ;
016060  023765  002330  000026          CMP     KW.EL,C.TO(R5)          ;
016066  103757                          BLO     1$                      ;
016070  016402  000002          2$:     MOV     2(R4),R2                ; GET REGISTER CONTENTS
016074                                  ERRDF   40,,ERR040              ; REPORT TIME OUT ERROR
016074  104455                          TRAP    C$ERDF
016076  000050                          .WORD   40
016100  000000                          .WORD   0
016102  006752                          .WORD   ERR040
016104  000407                          BR      4$                      ;
                                        ;
                                        ;       CHECK IF ERROR BIT SET
                                        ;
016106  016402  000002          3$:     MOV     2(R4),R2                ; GET REGISTER CONTENTS
016112  100006                          BPL     5$                      ; EXIT IF ERROR NOT SET
016114                                  ERRDF   40,,ERR040              ; REPORT ERROR INFO
016114  104455                          TRAP    C$ERDF
016116  000050                          .WORD   40
016120  000000                          .WORD   0
016122  006752                          .WORD   ERR040
016124  000261          4$:     SEC                                     ;
016126  000207                  RETURN                                  ;
```

```
                                        ;
                                        ;       NORMAL EXIT
                                        ;
016130  000241                  5$:     CLC                             ; CLEAR CARRY AS NO ERROR INDICATION
016132  000207                          RETURN                          ;
```

```
                                        ;RESET
                                        ;
                                        ; RESET ALL CONTROLLERS IN THE CONTROLLER TABLES
                                        ;
                                        ; INPUTS:
                                        ;       IPADRS - CONTAINS ALL IP ADDRESSES
                                        ; OUTPUTS:
                                        ;       NONE
                                        ;

016134  005037  002340          RESET:  CLR     NXMAD                   ; CLEAR NON-EXISTANT MEMORY ADDRESS
016140                                  PUSH    <R3,R4>                 ; SAVE R3 AND R4 ON STACK
016144                                  SETVEC  @ERRVEC,@NXMI,@PRIO7    ; SETUP TIMEOUT ERROR VECTOR
016144  012746  000340                  MOV     @PRIO7,-(SP)
016150  012746  015146                  MOV     @NXMI,-(SP)
016154  012746  000004                  MOV     @ERRVEC,-(SP)
016160  012746  000003                  MOV     #3,-(SP)
016164  104437                          TRAP    C$SVEC
016166  062706  000010                  ADD     #10,SP
016172  012703  000004                  MOV     #4.,R3                  ; STORE MAXIMUM # OF CONTROLLERS IN R3
016176  012704  002244                  MOV     #IPADRS,R4              ; STORE IP ADDRESS IN R4
016202  005714                  1$:     TST     (R4)                    ; IS THERE AN ENTRY?
016204  001403                          BEQ     2$                      ; IF NOT, DONE
016206  005034                          CLR     @(R4)+                  ; INIT CONTROLLER
016210  005303                          DEC     R3                      ; MAKE SURE WE DO NOT EXTEND OVER AREA
016212  001373                          BNE     1$                      ; IF NOT DONE, BRANCH
016214  005737  002320          2$:     TST     KW.CSR                  ; SEE IF CLOCK PRESENT,
016220  001403                          BEQ     3$                      ; BRANCH IF NOT, ELSE
016222  012777  000105  164070          MOV     #KW.OUT,@KW.CSR         ; START THE CLOCK.
016230                          3$:     POP     <R4,R3>                 ; RESTORE R3,R4 FROM STACK
016234  000207                          RETURN                          ;
```

```
 1
 2                                      ;RNTIME
 3                                      ;
 4                                      ;PRINT RUNTIME
 5                                      ;
 6                                      ;INPUTS:
 7                                      ;       KW.EL - CONTAINS ELAPSED TIME
 8                                      ;       KW.HZ - HERTZ OF CLOCK
 9                                      ;OUTPUTS:
10                                      ;       IF CLOCK ON SYSTEM:
11                                      ;          " RUN TIME HH:MM:SS " PRINTED
12                                      ;       IF NO CLOCK: ONE SPACE IS PRINTED
13
14 016236  005737  002320      RNTIME: TST     KW.CSR                  ;CHECK IF A CLOCK PRESENT
15 016242  001465                      BEQ     RNTIMX                  ;BRANCH IF NOT
16 016244                              PUSH    <R0,R3,R4,R5>
17 016254  013703  002330              MOV     KW.EL,R3                        ;GET ELAPSED TIME
18 016260  013704  002332              MOV     KW.EL+2,R4
19 016264  013700  002326              MOV     KW.HZ,R0                        ;GET SPEED OF CLOCK
20 016270  004737  013634              CALL    DIVIDE                  ;COMPUTE SECONDS OF ELAPSED TIME
21 016274  012700  000074              MOV     #60.,R0                 ;NOW DIVIDE BY 60
22 016300  004737  013634              CALL    DIVIDE                  ; TO COMPUTE MINUTES
23 016304                              PUSH    R5                              ;SAVE REMAINDER AS SECONDS
24 016306  004737  013634              CALL    DIVIDE                  ;DIVIDE BY 60 AGAIN
25 016312                              PNT     RNTIM,R3                        ;PRINT HOURS
26 016324  020527  000011              CMP     R5,#9.                  ;IF MINUTES 9 OR LESS
27 016330  003004                      BGT 1$
28 016332                              PRINT   #'0                     ;PRINT A LEADING ZERO
29 016342                      1$:     PNT     RNTIM1,R5                       ;NOW PRINT MINUTES
30 016354                              POP     R5                              ;GET SECONDS
31 016356  020527  000011              CMP     R5,#9.                  ;IF 9 OR LESS
32 016362  003004                      BGT 2$
33 016364                              PRINT   #'0                     ;PRINT A LEADING ZERO
34 016374                      2$:     PNT     RNTIM2,R5                       ;NOW PRINT SECONDS
35 016406                              POP     <R5,R4,R3,R0>           ;HOURS IN R3
36 016416                      RNTIMX: PRINT   <#' >                   ;PRINT A SPACE
37 016426  000207                      RETURN
65
```

```
                                .SBTTL  REPORT CODING SECTION

                                ;++
                                ;       THE REPORT CODING SECTION CONTAINS THE CODE FOR PRINTING
                                ;       STATISTICAL INFORMATION GATHERED BY THE DIAGNOSTIC.  IT IS
                                ;       EXECUTED BY THE OPERATOR COMMAND "PRINT" OR BY THE MACRO CALL
                                ;       "DORPT".
                                ;--

016430                                  BGNRPT
016430                          L$RPT::

016430                                  PUSH    <R0,R1,R2,R3,R4,R5>
016444                                  PNTS    RPTMSG,TNUM             ; PRINT TEST NUMBER
016460  004737  016236                  CALL    RNTIME                  ; GET RUNTIME PARAMETERS
016464                                  PRINT   #CR                     ; END THE LINE
016474  012701  002334                  MOV     #STIME,R1               ; GET REPORT TIMER
016500  012700  001604                  MOV     #15.*60.,R0             ; GET REPORT INTERVAL
016504  004737  015210                  CALL    SETTO                   ; SET TIME FOR NEXT REPORT

016510                          1$:     PNTS    RPTMSH
016520  013705  002170                  MOV     CTABS,R5                ;GET ADDRESS OF 1ST CONTROLLER TABLE

016524  005765  000002          RPTCT:  TST     C.UNIT(R5)              ; SEE IF CONTROLLER IS AVAILABLE
016530  100010                          BPL     1$                      ; IF SO, SKIP (BIT 15 = 0)
016532                                  PRINTS  #RPTMS5                 ;
016532  012746  017702                  MOV     #RPTMS5,-(SP)
016536  012746  000001                  MOV     #1,-(SP)
016542  010600                          MOV     SP,R0
016544  104416                          TRAP    C$PNTS
016546  062706  000004                  ADD     #4,SP
016552                          1$:
016552  010504                          MOV     R5,R4                   ; GET ADDRESS OF CONTROLLER TABLE
016554  062704  000016                  ADD     #C.DRO,R4               ; POINT TO DRIVE TABLE POINTERS
016560  012703  000004                  MOV     #4.,R3                  ; GET COUNT OF DRIVES
016564  012401                  RPTDT:  MOV     (R4)+,R1                ; LOOK AT POINTER
016566  001571                          BEQ     RPTCTN                  ; GO TO NEXT IF NO TABLE
016570  005761  000002                  TST     D.UNIT(R1)              ; SEE IF DRIVE AVAILABLE
016574  100010                          BPL     5$                      ; IF SO, SKIP (BIT 15 = 0)
016576                                  ASSUME DT.AVL EQ BIT15
016576                                  PRINTS  #RPTMS4
016576  012746  017631                  MOV     #RPTMS4,-(SP)
016602  012746  000001                  MOV     #1,-(SP)
016606  010600                          MOV     SP,R0
016610  104416                          TRAP    C$PNTS
016612  062706  000004                  ADD     #4,SP
016616                          5$:
016616                                  PUSH    <R3,R4,R5,R1>
016626  012700  002214                  MOV     #TEMP,R0                ;PLACE 18 SPACE CHARACTERS INTO
016632  012701  000022                  MOV     #18.,R1                 ;  TEMP STORAGE
016636  112720  000040          1$:     MOVB    #' ,(R0)+
 16642  005301                          DEC     R1
016644  001374                          BNE     1$
016646  005010                          CLR     (R0)                    ;THEN A NULL CHARACTER
016650  011605                          MOV     (SP),R5                 ;GET DRIVE TABLE STORAGE ADDRESS
016652  016501  000200                  MOV     D.SERN(R5),R1           ;GET SERIAL NUMBER
016656  016502  000202                  MOV     D.SERN+2(R5),R2
```

```
     48 016662  016503  000204                    MOV     D.SERN+4(R5),R3
     49 016666  005004                            CLR     R4
     50 016670  004737  013672          2$:       CALL    DIV10                   ;DIVIDE BY 10
     51 016674  062705  000060                    ADD     #'0,R5                  ;CONVERT TO ASCII CHARACTER
     52 016700  110540                            MOVB    R5,-(R0)                ;PUT DIGIT INTO TEMP STORAGE
     53 016702  010146                            MOV     R1,-(SP)
     54 016704  050216                            BIS     R2,(SP)                 ;SEE IF QUOTIENT IS ZERO
     55 016706  050316                            BIS     R3,(SP)
     56 016710  050426                            BIS     R4,(SP)+
     57 016712  001366                            BNE     2$                      ;IF NOT, DIVIDE AGAIN
     58 016714                                    POP     R1
     59 016716                                    PRINTS  #RPTMSD,D.UNIT(R1),(R1),#TEMP,D.SEEK(R1),D.XFRR(R1),D.XFRW(R1)
        016716  016146  000164                    MOV     D.XFRW(R1),-(SP)
        016722  016146  000166                    MOV     D.XFRR(R1),-(SP)
        016726  016146  000174                    MOV     D.SEEK(R1),-(SP)
        016732  012746  002214                    MOV     #TEMP,-(SP)
        016736  011146                            MOV     (R1),-(SP)
        016740  016146  000002                    MOV     D.UNIT(R1),-(SP)
        016744  012746  017460                    MOV     #RPTMSD,-(SP)
        016750  012746  000007                    MOV     #7,-(SP)
        016754  010600                            MOV     SP,R0
        016756  104416                            TRAP    C$PNTS
        016760  062706  000020                    ADD     #20,SP
     60 016764                                    ASSUME D.DRV EQ 0
     61 016764                                    PRINTS  #RPTMD2,D.HERR(R1),D.SERR(R1),D.ECCC(R1)
        016764  016146  000176                    MOV     D.ECCC(R1),-(SP)
        016770  016146  000172                    MOV     D.SERR(R1),-(SP)
        016774  016146  000170                    MOV     D.HERR(R1),-(SP)
        017000  012746  017527                    MOV     #RPTMD2,-(SP)
        017004  012746  000004                    MOV     #4,-(SP)
        017010  010600                            MOV     SP,R0
        017012  104416                            TRAP    C$PNTS
        017014  062706  000012                    ADD     #12,SP
     62 017020                                    PUSH    R1
     63 017022  016102  000212                    MOV     D.HDAS+2(R1),R2
     64 017026  016103  000214                    MOV     D.HDAS+4(R1),R3
     65 017032  016104  000216                    MOV     D.HDAS+6(R1),R4
     66 017036  016101  000210                    MOV     D.HDAS(R1),R1
     67 017042  012700  002214                    MOV     #TEMP,R0
     68 017046  012705  000024                    MOV     #20.,R5
     69 017052  112720  000040          3$:       MOVB    #' ,(R0)+
     70 017056  005305                            DEC     R5
     71 017060  001374                            BNE     3$
     72 017062  005010                            CLR     (R0)
     73 017064  004737  013672          4$:       CALL    DIV10
     74 017070  062705  000060                    ADD     #'0,R5
     75 017074  110540                            MOVB    R5,-(R0)
     76 017076  010146                            MOV     R1,-(SP)
     77 017100  050216                            BIS     R2,(SP)
     78 017102  050316                            BIS     R3,(SP)
     79 017104  050426                            BIS     R4,(SP)+
     80 017106  001366                            BNE     4$
     81 017110                                    POP     R1
     82 017112                                    PRINTS  #RPTMD3,R0,D.SKER(R1)
        017112  016146  000206                    MOV     D.SKER(R1),-(SP)
        017116  010046                            MOV     R0,-(SP)
        017120  012746  017551                    MOV     #RPTMD3,-(SP)
```

```
                017124  012746  000003                  MOV     #3,-(SP)
                017130  010600                          MOV     SP,R0
                017132  104416                          TRAP    C$PNTS
                017134  062706  000010                  ADD     #10,SP
                                                        POP     <R5,R4,R3>
     83         017140
     84         017146  005303                  RPTDTN: DEC     R3                      ;COUNT THE DRIVE TABLES
     85         017150  003205                          BGT RPTDT                       ;REPEAT FOR ALL DRIVE TABLES
     86         017152  062705  000034          RPTCTN: ADD     #C.SIZE,R5              ;GO TO NEXT CONTROLLER TABLE
     87         017156  005715                          TST     (R5)
     88         017160  001402                          BEQ     RPTXX
     89         017162  000137  016524                  JMP     RPTCT
     91         017166                          RPTXX:  POP     <R5,R4,R3,R2,R1,R0>
     92
     93         017172                                  EXIT    RPT
                017174  000167                          .WORD   J$JMP
                017176  000554                          .WORD   L10023-2-.
     94
     95         017200      116     042     124  RPTMSG: .ASCIZ \N"TEST "D3" IN PROGRESS.  "\
     97         017242      116     042     125  RPTMSH: .ASCII \N"UNIT DRIVE      SERIAL-NUMBER SEEKS MBYTES  MBYTES  HARD   SOFT    ECC"N\
     98         017354      042     040     040          .ASCIZ \"                                    X1000  READ  WRITTEN ERRORS ERRORS"N\
     99         017460      045     123     062  RPTMSD: .ASCIZ \%S2%D2%S3%D3%S1%T%S1%D5%S2%D5%S3%D5%S2\
    100         017527      045     104     065  RPTMD2: .ASCIZ \%D5%S2%D5%S1%D5%N\
    101         017551      045     101     011  RPTMD3: .ASCIZ \%A      HDA SERIAL NUMBER %T%A     SEEK ERRORS %D5%N\
    102         017631      045     101     040  RPTMS4: .ASCIZ \%A *** FOLLOWING UNIT WAS DROPPED *** %N\
    103         017702      045     101     040  RPTMS5: .ASCIZ \%A *** FOLLOWING CONTROLLER WAS DROPPED *** %N\
    105                                              .EVEN
    106
    107         017762                                  ENDRPT
                017762                          L10023:
                017762  104425                          TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

                                ;++
                                ;       THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ;       TO PROTECT THE LOAD MEDIA.
                                ;--

                                        BGNPROT
017764                          L$PROT::

017764  177777                          -1                      ; P-TABLE OFFSET FOR CSR ADDRESS
017766  177777                          -1                      ; P-TABLE OFFSET FOR MASSBUS ADDRESS
017770  177777                          -1                      ; P-TABLE OFFSET FOR DRIVE NUMBER

017772                                  ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

                                ;++
                                ;       THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;       AT THE BEGINNING OF EACH PASS. THIS CODE IS EXECUTED UNDER FIVE
                                ;       CONDITIONS.  THERE
                                ;       ARE SUPERVISOR EVENT FLAGS THAT ARE USED TO LET THE
                                ;       DIAGNOSTIC KNOW UNDER WHICH CONDITION THE EXECUTION IS TAKING
                                ;       PLACE.  THE EVENT FLAGS ARE READ USING THE "READEF" MACRO.
                                ;       THE CONDITIONS UNDER WHICH THE INIT CODE IS EXECUTED AND THE
                                ;       CORRRESPONDING EVENT FLAGS ARE:
                                ;               START COMMAND           EF.START
                                ;               RESTART COMMAND         EF.RESTART
                                ;               CONTINUE COMMAND        EF.CONTINUE
                                ;               POWERDOWN/POWERUP       EF.PWR
                                ;               NEW PASS                EF.NEW
                                ;
                                ;       IF HERE FROM START COMMAND THEN
                                ;               SET ISTRT BIT & CLEAR OTHER BITS IN FLAG
                                ;
                                ;       IF HERE FROM RESTART COMMAND THEN
                                ;               SET IREST BIT IN IFLAGS
                                ;
                                ;       IF HERE FROM START OR RESTART COMMAND THEN
                                ;               RESET ALL UNITS
                                ;               ESTABLISH FREE MEMORY
                                ;               CLEAR TNUM
                                ;               INITIALIZE CLOCK
                                ;               BUILD CONTROLLER & DRIVES TABLES IN MEMORY
                                ;               EXIT INIT SECTION
                                ;
                                ;       IF HERE FROM CONTINUE COMMAND THEN
                                ;               SET ICONT BIT IN IFLAGS
                                ;               EXIT INIT SECTION
                                ;
                                ;       IF HERE FROM POWER FAIL RESTART THEN
                                ;               EXIT INIT SECTION
                                ;
                                ;       IF HERE FROM NEW PASS OR SUB-PASS THEN
                                ;               LOOK FOR ANY ADDED OR DROPPED UNITS
                                ;               EXIT INIT SECTION
                                ;--

                                        BGNINIT
017772                          L$INIT::
                                                                        ;HERE FROM START COMMAND?
017772                                  READEF  #EF.STA
017772  012700  000040                  MOV     #EF.STA,R0
017776  104447                          TRAP    C$REFG
                                                                        ;BRANCH TO 1$ IF NOT, ELSE
                                        BNCOMPLETE      1$
020000  103004                          BCC     1$
020002  012737  000010  002200          MOV     #ISTRT,IFLAGS           ;SET START BIT IN FLAG.
020010  000525                          BR      INIT1
020012                          1$:                                     ;HERE FROM RESTART COMMAND?
020012                                  READEF  #EF.RES
020012  012700  000037                  MOV     #EF.RES,R0
```

INITIALIZE SECTION

```
                020016  104447                          TRAP    C$REFG
                                                                                ;BRANCH TO 2$ IF NOT, ELSE
                53
                54 020020
                020020  103004                          BCC     2$
                55 020022  052737  000004  002200        BIS     #IREST,IFLAGS           ;SET RESTART BIT IN FLAG.
                56 020030  000515                        BR      INIT1
                57 020032                       2$:                                     ;HERE FROM CONTINUE COMMAND?
                58 020032                               READEF  #EF.CON
                020032  012700  000036                   MOV     #EF.CON,R0
                020036  104447                          TRAP    C$REFG
                                                                                ;BRANCH TO 3$ IF NOT, ELSE
                59
                60 020040                               BNCOMPLETE      3$
                020040  103007                          BCC     3$
                61 020042  042737  000020  002200        BIC     #ISTRTH,IFLAGS          ;CLEAR 1ST TIME THRU CZUDIO FLAG AND
                62 020050  052737  000002  002200        BIS     #ICONT,IFLAGS           ;SET CONTINUE BIT IN FLAG.
                63 020056  000472                        BR      13$
                64 020060                       3$:                                     ;HERE FROM POWER FAIL?
                65 020060                               READEF  #EF.PWR
                020060  012700  000034                   MOV     #EF.PWR,R0
                020064  104447                          TRAP    C$REFG
                                                                                ;BRANCH TO 4$ IF NOT, ELSE
                66
                67 020066                               BNCOMPLETE      4$
                020066  103001                          BCC     4$
                68 020070  000465                        BR      13$
                69
                70                              ;
                71                              ;       MAKE ALL CONTROLLER/DRIVE TABLES NOT AVAILABLE FOR TESTING
                72                              ;
                73
                74 020072  013705  002170        4$:     MOV     CTABS,R5                ;GET ADDRESS OF 1ST CONTROLLER TABLE
                75 020076  052765  100000  000002  5$:     BIS     #CT.AVL,C.UNIT(R5)      ;SET CONTROLLER TABLE NOT AVAILABLE
                76 020104  010502                        MOV     R5,R2                   ;GET POINTER TO DRIVE TABLES
                77 020106  062702  000016                ADD     #C.DRO,R2
                78 020112  012703  000004                MOV     #4.,R3                  ; GET NUMBER OF DRIVES PER CONTROLLER
                79 020116  012200                6$:     MOV     (R2)+,R0                ;SEE IF THIS DRIVES HAS A TABLE.
                80 020120  001403                        BEQ     7$                      ;BRANCH IF NOT, ELSE
                81 020122  052760  100000  000002        BIS     #DT.AVL,D.UNIT(R0)      ;SET DRIVE TABLE NOT AVAILABLE.
                82 020130  005303                7$:     DEC     R3                      ;LOOK AT NEXT DRIVE IN CONTROLLER TABLE.
                83 020132  001371                        BNE     6$                      ;BRANCH IF NO DRIVES, ELSE
                84 020134  062705  000034                ADD     #C.SIZE,R5              ;MOVE TO NEXT CONTROLLER TABLE
                85 020140  005715                        TST     (R5)                    ;IS THERE A NEXT ONE?
                86 020142  001355                        BNE     5$                      ;IF SO, CLEAR THE BITS THERE
                87                              ;
                88                              ;       NOW GET EACH P-TABLE AND MAKE THE APPROPRIATE CONTROLLER/DRIVE
                89                              ;       TABLES AVAILABLE FOR TESTING.
                90                              ;
                91 020144  005003                        CLR     R3                      ;START WITH LOGICAL UNIT 0
                92 020146                       8$:                                     ;GET POINTER TO IT'S P-TABLE
                93 020146                               GPHARD  R3,R0
                020146  010300                          MOV     R3,R0
                020150  104442                          TRAP    C$GPHRD
                                                                                ;BRANCH TO 12$ IF NOT AVAILABLE
                94
                95 020152                               BNCOMPLETE      12$
                020152  103030                          BCC     12$
                96 020154  013705  002170                MOV     CTABS,R5                ;GET ADDRESS OF 1ST CONTROLLER TABLE
                97 020160  021015                9$:     CMP     (R0),(R5)               ;SEE IF CSR ADDRESSES ARE THE SAME.
                98 020162  001411                        BEQ     11$                     ;BRANCH IF SO, ELSE
```

INITIALIZE SECTION

```
 99 020164  062705  000034                    ADD     #C.SIZE,R5              ;LOOK AT NEXT CONTROLLER TABLE.
100 020170  005715                            TST     (R5)                    ;SEE IF THERE IS ANOTHER CONTROLLER TABLE,
101 020172  001372                            BNE     9$                      ;BRANCH IF SO, ELSE
102 020174                              10$:                                  ;REPORT TABLE CONSISTANCY ERROR.
103 020174                                    ERRSF   6.,ERR006
    020174  104454                           TRAP    C$ERSF
    020176  000006                           .WORD   6
    020200  000000                           .WORD   0
    020202  006606                           .WORD   ERR006
104                                                                           ;DO CLEAN-UP TRAP
105 020204                                    DOCLN
    020204  104444                           TRAP    C$DCLN
106
107 020206  016001  000002              11$:  MOV     H.DRV(R0),R1            ;GET DRIVE NUMBER FROM P-TABLE
108 020212  004737  013224                    CALL    GTDRVT                  ;FIND THE DRIVE TABLE ADDRESS
109 020216  001366                            BNE     10$                     ;BRANCH IF NOT FOUND, ELSE
110 020220  042765  100000  000002            BIC     #CT.AVL,C.UNIT(R5)      ;CLEAR AVAILABLE BIT IN CONTROLLER AND
111 020226  042764  100000  000002            BIC     #DT.AVL,D.UNIT(R4)      ;THE DRIVE TABLES.
112 020234  005203                      12$:  INC     R3                      ;INCREMENT TO NEXT UNIT IN P-TABLE
113 020236  020337  002012                    CMP     R3,L$UNIT               ;SEE IF ALL P-TABLES CHECKED,
114 020242  002741                            BLT     8$                      ;BRANCH IF NOT, ELSE
115 020244  012701  002334              13$:  MOV     #STIME,R1               ; GET REPORT TIMER
116 020250  012700  001604                    MOV     #15.*60.,R0             ; GET REPORT INTERVAL
117 020254  004737  015210                    CALL    SETTO                   ; SET TIME FOR NEXT REPORT
118 020260  000137  021232                    JMP     INITXX                  ;EXIT THE INITIALIZE SECTION.
119
120                                     ;
121                                     ;     INITIALIZE KW11 CLOCK, FREE MEMORY AND IP ADDRESS TABLE
122                                     ;     DURING START OR RESTART COMMAND ONLY
123                                     ;
124
125 020264  005037  002330              INIT1:CLR     KW.EL                   ;CLEAR ELAPSED TIME
126 020270  005037  002332                    CLR     KW.EL+2
127 020274                                    CLOCK   L,R0                    ;SEE IF L-CLOCK PRESENT
    020274  012700  000114                    MOV     #'L,R0
    020300  104462                           TRAP    C$CLCK
128 020302                                    BCOMPLETE       1$
    020302  103413                           BCS     1$
129 020304                                    CLOCK   P,R0                    ;SEE IF P-CLOCK PRESENT
    020304  012700  000120                    MOV     #'P,R0
    020310  104462                           TRAP    C$CLCK
130 020312                                    BCOMPLETE       1$
    020312  103407                           BCS     1$
131 020314  005037  002320                    CLR     KW.CSR                  ;IF NEITHER, CLEAR CSR STORAGE WORD
132 020320                                    PNTF    NOCLOCK
133 020330  000434                           BR      2$
134
135 020332  012037  002320              1$:   MOV     (R0)+,KW.CSR            ;STORE DATA RETURNED
136 020336  012037  002322                    MOV     (R0)+,KW.BRL
137 020342  012037  002324                    MOV     (R0)+,KW.VEC
138 020346  012037  002326                    MOV     (R0)+,KW.HZ
139                                                                           ;SETUP KW11 VECTOR ADDRESS
140 020352                                    SETVEC  KW.VEC,#KW11I,#PRI07
    020352  012746  000340                    MOV     #PRI07,-(SP)
    020356  012746  015166                    MOV     #KW11I,-(SP)
    020362  013746  002324                    MOV     KW.VEC,-(SP)
    020366  012746  000003                    MOV     #3,-(SP)
```

```
                020372  104437                          TRAP    C$SVEC
                020374  062706  000010                  ADD     #10,SP
    141         020400  012777  000105  161712          MOV     #KW.OUT,@KW.CSR         ;START THE CLOCK
    142         020406  012701  002334                  MOV     #STIME,R1               ; GET REPORT TIMER
    143         020412  012700  001604                  MOV     #15.*60.,R0             ; GET REPORT INTERVAL
    144         020416  004737  015210                  CALL    SETTO                   ; SET TIME FOR NEXT REPORT
    145         020422  004737  016134          2$:     CALL    RESET                   ;RESET ALL CONTROLLERS
    146         020426                                  MEMORY  FFREE                   ;RESET START OF FREE MEMORY
                020426  104431                          TRAP    C$MEM
                020430  010037  002156                  MOV     R0,FFREE
    147         020434  017737  161516  002160          MOV     @FFREE,FSIZE            ;RESET SIZE OF FREE MEMORY
    148         020442  005037  002202                  CLR     TNUM                    ;INITIALIZE TEST NUMBER TO NO TEST RUNNING
    149
    150                                         ;
    151                                         ;       ALLOCATE DRIVE TABLES TO MEMORY
    152                                         ;
    153
    154         020446  013737  002156  002166  INIT2:  MOV     FFREE,DTABS             ;STORE START OF DRIVE TABLES AND
    155         020454  005077  161506                  CLR     @DTABS                  ;MARK ZERO END.
    156         020460  013700  002012                  MOV     L$UNIT,R0               ;GET NUMBER OF LOGICAL UNITS TO RUN,
    157         020464  012701  000001                  MOV     #1,R1                   ;GET INITIAL SIZE OF DRIVE TABLE AND
    158         020470  062701  000110          1$:     ADD     #<D.SIZE>/2,R1          ;ACCUMULATE DRIVE TABLE SIZE.
    159         020474  005300                          DEC     R0                      ;SEE IF ANY MORE LOGICAL UNITS,
    160         020476  001374                          BNE     1$                      ;BRANCH IF NOT, ELSE
    161         020500  004737  007332                  CALL    ALOCM                   ;ALLOCATE ALL DRIVE TABLES TO MEMORY.
    162                                                                                 ; R1 POINTS TO 1ST WORD IN DRIVE TABLE
    163
    164                                         ;
    165                                         ;       INITIALIZE CONTROLLER TABLE STORAGE WITH A WORD OF ZEROS
    166                                         ;
    167
    168         020504  013737  002156  002170  INIT3:  MOV     FFREE,CTABS             ; STORE START OF CONTROLLER TABLES AND
    169         020512  005077  161452                  CLR     @CTABS                  ; MARK ZEROS END.
    170         020516  005037  002172                  CLR     CTRLRS                  ; CLEAR CONTROLLER COUNT
    171         020522  012701  002244                  MOV     #IPADRS,R1              ; R1 -> IP ADDRESS
    172         020526  012702  000004                  MOV     #4.,R2                  ; GET MAXIMUM # OF CONTROLLERS
    173         020532  005021                  1$:     CLR     (R1)+                   ; CLEAR ENTRY
    174         020534  005302                          DEC     R2                      ; DONE?
    175         020536  001375                          BNE     1$                      ; IF NOT, BRANCH
    176
    177                                         ;
    178                                         ;       BUILD CONTROLLER TABLES
    179                                         ;
    180
    181         020540  005005          INIT4:  CLR     R5                      ;CLEAR CUSTOMER DATA FLAG
    182         020542  005002                  CLR     R2                      ;START WITH LOGICAL UNIT 0
    208         020544  012737  005160  002254          MOV     #5160,BRSAV             ; SAVE DEFAULT FOR BR LEVEL & VECTOR
    209         020552                  1$:     GPHARD  R2,R0                   ; GET POINTER TO IT'S P-TABLE
                020552  010200                  MOV     R2,R0
                020554  104442                  TRAP    C$GPHRD
    210         020556                          BNCOMPLETE      16$             ; BRANCH TO 16$ IF NOT AVAILABLE
                020556  103124                  BCC     16$
    211         020560  013703  002170          MOV     CTABS,R3                ; GET ADDRESS OF 1ST CONTROLLER TABLE
    212         020564  005713          2$:     TST     (R3)                    ; CHECK IF ANY MORE TABLES
    213         020566  001405                  BEQ     6$                      ; BUILD NEW TABLE IF FOUND ZERO WORD
    214         020570  021013                  CMP     (R0),(R3)               ; CHECK IF SAME CSR ADDRESS.
    215         020572                                  ASSUME  C.UADR EQ 0
```

```
                                                ASSUME  H.UBA EQ 0
020572
020572  001444                          BEQ     11$                     ; BRANCH IF SO

020574  062703  000034          5$:     ADD     #C.SIZE,R3              ;POINT TO BEGINNING OF NEXT CONTROLLER
020600  000771                          BR      2$                      ;TABLE IN MEMORY.

                                ;
                                ;       BUILD NEW CONTROLLER TABLE
                                ;

020602  012704  002244          6$:     MOV     #IPADRS,R4              ;GET BEGINNING OF IP ADDRESS TABLE
020606  020427  002254          7$:     CMP     R4,#IPADRS+8.           ;SEE IF END OF IP ADDRESS TABLE.
020612  101004                          BHI     9$                      ;BRANCH IF SO, ELSE
020614  005724                          TST     (R4)+                   ;DID WE FIND AN OPEN ENTRY ?
020616  001401                          BEQ     8$                      ;BRANCH IF SO, ELSE
020620  000772                          BR      7$                      ;LOOK AGAIN.

020622  011044                  8$:     MOV     (R0),-(R4)              ;TAKE CSR ADDRESS FROM P-TABLE
                                                                        ;AND STORE IT IN THE IP ADDRESS TABLE.
020624  012701  000016          9$:     MOV     #<C.SIZE>/2,R1          ;GET # OF ENTRIES IN CONTROLLER TABLE
020630  004737  007332                  CALL    ALOCM                   ;AND ALLOCATE A TABLE TO MEMORY.
                                                                        ; R0 => 1ST WORD P-TABLE
                                                                        ; R1 => 1ST WORD IN CONTROLLER TABLE
020634  011021                          MOV     (R0),(R1)+              ; STORE CSR ADDRESS AND
020636  010221                          MOV     R2,(R1)+                ; UNIT NUMBER IN THE CONTROLLER TABLE.
020640  013704  002254                  MOV     BRSAV,R4                ; GET DEFAULT VECTOR & BR LEVEL
020644  162704  000004                  SUB     #4,R4                   ; GET NEXT VECTOR
020650  010437  002254                  MOV     R4,BRSAV                ; SAVE NEXT VECTOR
020654  010421                          MOV     R4,(R1)+                ;STORE IT IN THE CONTROLLER TABLE.
020656  012721  004037                  MOV     #4037,(R1)+             ;THE 'JSR R0' INSTRUCION AND
020662  012721  015156                  MOV     #CNTSRV,(R1)+           ;THE ADDRESS OF THE INTERRUPT SERVICE
                                                                        ;ROUTINE IN THE CONTROLLER TABLE.
020666  012704  000011                  MOV     #<C.SIZE-C.FLG>/2,R4    ;GET # OF ENTRIES TO END OF TABLE.
020672  005021          10$:    CLR     (R1)+                   ;CLEAR REST OF TABLE AND
020674  005304                  DEC     R4                      ;ADD ZERO WORD AT END.
020676  002375                  BGE     10$                     ;LOOP TIL ALL CLEARED
020700  005237  002172                  INC     CTRLRS                  ;KEEP TRACK OF CONTROLLER COUNT

                                ;
                                ;       BUILD DRIVE TABLES
                                ;

020704  013701  002166          11$:    MOV     DTABS,R1                ;GET ADDRESS OF CURRENT DRIVE TABLE
020710  062703  000016                  ADD     #C.DR0,R3               ; INDEX TO 1ST DRIVE IN TABLE
020714  012704  000004                  MOV     #4.,R4                  ; GET # OF DRIVES PER CONTROLLER
020720  005713          12$:    TST     (R3)                    ;ANY ENTRY TO DRIVE TABLE.
020722  001411                  BEQ     14$                     ;BRANCH IF NOT, ELSE
020724  026033  000002                  CMP     H.DRV(R0),0(R3)+        ;COMPARE DRIVE NUMBER IN DRIVE TABLE.
020730  001002                  BNE     13$                     ;BRANCH IF DIFFERNENT, ELSE
020732  000137  021244                  JMP     MLDRER                  ;FOUND TWO P-TABLES WITH SAME DRIVE.

020736  005304          13$:    DEC     R4                      ; COUNT DRIVES
020740  001367                  BNE     12$                     ; IF FOUR DRIVE TABLES ALREADY EXIST,
020742  000137  021262                  JMP     TOOMER                  ;  THEN REPORT ERROR

020746  010113          14$:    MOV     R1,(R3)                 ;STORE ADDRESS OF DRIVE TABLE IN
                                                                ;CONTROLLER TABLE.
```

INITIALIZE SECTION

```
281 020750  016021  000002                      MOV     H.DRV(R0),(R1)+         ;STORE DRIVE NUMBER AND
282 020754  010221                              MOV     R2,(R1)+                ;LOGICAL UNIT NUMBER IN DRIVE TABLE.
284 020756  016011  000004                      MOV     H.PRM(R0),(R1)          ;GET TEST AREA BIT
285 020762  051105                              BIS     (R1),R5                 ;SAVE "OR" OF BIT FROM ALL DRIVES
286 020764  005111                              COM     (R1)                    ;COMPLIMENT IT
287 020766  042711  157777                      BIC     #^C<HM.CYL>,(R1)        ; SET BIT FOR WRITE ON CUST DATA
288 020772  052721  011012                      BIS     #DDEF,(R1)+             ;LOAD DEFAULT PARAMETER BITS
289 020776  012704  000105                      MOV     #<D.SIZE/2>-3,R4        ;CLEAR REST OF TABLE
290 021002  005021                  15$:        CLR     (R1)+
291 021004  005304                              DEC     R4
292 021006  003375                              BGT     15$
293 021010  012761  177777  177742              MOV     #-1,<D.ECYL+2-D.SIZE>(R1) ;MARK CYLINDERS AT TEST ALL
295
296 021016  062737  000220  002166              ADD     #D.SIZE,DTABS           ;NEXT DRIVE TABLE ADDRESS AND
297 021024  005077  161136                      CLR     @DTABS                  ;MARK ZERO END.
298 021030  005202                  16$:        INC     R2                      ;INCREMENT LOGICAL UNIT NUMBER
299 021032  020237  002012                      CMP     R2,L$UNIT               ;CHECK IF GOT ALL TABLES
300 021036  002645                              BLT     1$                      ;IF NOT, GO BACK FOR NEXT, ELSE
301 021040  012701  000001                      MOV     #1,R1                   ;GET 1 WORD TO TERMINATE ALL CONTROLLER
302 021044  004737  007332                      CALL    ALOCM                   ;TABLES AND ALLOCATE IT TO MEMORY.
303
305                                     ;
306                                     ;       CHECK FOR CUSTOMER WARNING MESSAGE
307                                     ;
308
309 021050  032705  020000              INIT5:  BIT     #HM.CYL,R5              ;CHECK IF BIT EVER SET
310 021054  001460                              BEQ     INIT6                   ;BYPASS IF NOT
311 021056                                      PNTF    INITWA                  ;PRINT WARNING HEADER
312 021066  013705  002170                      MOV     CTABS,R5                ;GET ADDRESS 1ST CONTROLLER TABLE
313 021072  010504                  1$:         MOV     R5,R4                   ;GET ADDRESS OF POINTER TO DRIVE TABLE
314 021074  062704  000016                      ADD     #C.DRO,R4
315 021100  012701  000004                      MOV     #4.,R1                  ;GET COUNT OF DRIVE TABLES
316 021104  012403                  2$:         MOV     (R4)+,R3                ;GET ADDRESS OF DRIVE TABLE
317 021106  001422                              BEQ     4$
318 021110  032763  020000  000004              BIT     #D.DCY,D.PRM(R3)        ;CHECK IF CUSTOMER DATA SELECTED
319 021116  001014                              BNE     3$
320 021120                                      PRINTF  #INITWB,D.UNIT(R3),(R5),(R3) ;PRINT NUMBERS
    021120  011346                              MOV     (R3),-(SP)
    021122  011546                              MOV     (R5),-(SP)
    021124  016346  000002                      MOV     D.UNIT(R3),-(SP)
    021130  012746  003356                      MOV     #INITWB,-(SP)
    021134  012746  000004                      MOV     #4,-(SP)
    021140  010600                              MOV     SP,R0
    021142  104417                              TRAP    C$PNTF
    021144  062706  000012                      ADD     #12,SP
321 021150  005301                  3$:         DEC     R1                              ;COUNT THE DRIVE TABLES
322 021152  001354                              BNE     2$                      ;LOOK AT ALL OF THEM
323 021154  062705  000034              4$:     ADD     #C.SIZE,R5              ;MOVE TO NEXT CONTROLLER TABLE
324 021160  005715                              TST     (R5)                    ;SEE IF ANOTHER TABLE AND
325 021162  001343                              BNE     1$                      ; LOOK AT IT
326
327                                     ;
328                                     ;       GET CONFIRMATION TO PROCEED
329                                     ;
330
331 021164                                      MANUAL                          ;CHECK IF MANUAL INTERVENTION ALLOWED
    021164  104450                              TRAP    C$MANI
```

INITIALIZE SECTION

```
                                                BNCOMPLETE      INIT6           ;BRANCH IF NOT ALLOWED
    021166  103013                              BCC     INIT6
                                                GMANIL  INITWC,TEMP,1,NO        ;ASK OPERATOR
    021170  104443                              TRAP    C$GMAN
    021172  000404                              BR      10000$
    021174  002214                              .WORD   TEMP
    021176  000120                              .WORD   T$CODE
    021200  002465                              .WORD   INITWC
    021202  000001                              .WORD   1
    021204                              10000$:
    021204  032737  000001  002214              BIT     #1,TEMP                 ;LOOK AT RESPONSE
    021212  001001                              BNE     INIT6                   ;BRANCH IF YES WAS ANSWER
                                                                                ;DO CLEAN-UP TRAP
    021214                                      DOCLN
    021214  104444                              TRAP    C$DCLN

                                        ;
                                        ;       SAVE CURRENT PARAMETERS TO FREE MEMORY SO EACH TEST CAN USE ALL OF IT
                                        ;

    021216  013737  002156  002162      INIT6:  MOV     FFREE,FMEM              ;SAVE START ADDRESS
    021224  013737  002160  002164              MOV     FSIZE,FMEMS             ;SAVE SIZE

                                        ;
                                        ;       EXIT INITIALIZE SECTION
                                        ;

    021232                              INITXX: SETPRI  #PRI00                  ;SET RUNNING PRIORITY TO ZERO
    021232  012700  000000                      MOV     #PRI00,R0
    021236  104441                              TRAP    C$SPRI

    021240                                      EXIT    INIT
    021240  104432                              TRAP    C$EXIT
    021242  000036                              .WORD   L10025-.
```

```
                                .SBTTL  INITIALIZE ERRORS

                                ;
                                ;       TWO P-TABLES FOR SAME DRIVE
                                ;

021244  013705  002214  MLDRER: MOV     TEMP,R5                 ;GET CONTROLLER ADDRESS
021250                          ERRSF 2,,ERR002
021250  104454                 TRAP    C$ERSF
021252  000002                 .WORD   2
021254  000000                 .WORD   0
021256  006540                 .WORD   ERR002
                                                                ;DO CLEAN-UP TRAP
021260                          DOCLN
021260  104444                 TRAP    C$DCLN

                                ;
                                ;       MORE THAN FOUR DRIVES SELECTED ON ONE CONTROLLER
                                ;

021262  013705  002214  TOOMER: MOV     TEMP,R5                 ;GET CONTROLLER ADDRESS
021266                          ERRSF 3,,ERR003
021266  104454                 TRAP    C$ERSF
021270  000003                 .WORD   3
021272  000000                 .WORD   0
021274  006556                 .WORD   ERR003
                                                                ;DO CLEAN-UP TRAP
021276                          DOCLN
021276  104444                 TRAP    C$DCLN

                                ENDINIT
021300                  L10025:
021300  104411                 TRAP    C$INIT
```

AUTODROP SECTION

```
                                .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.
                                ;--

                                        BGNAUTO
021302                          L$AUTO::
                                        ENDAUTO
021302                          L10026:
021302  104461                          TRAP    C$AUTO
```

CLEANUP CODING SECTION

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;++
                                ;       THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;       AFTER EACH PASS AND AFTER THE PROGRAM IS INTERRUPTED BY "↑C".
                                ;--

021304                                  BGNCLN
021304                          L$CLEAN::

021304                                  EXIT    CLN
021304  104432                          TRAP    C$EXIT
021306  000002                          .WORD   L10027-.

021310                                  ENDCLN
021310                          L10027:
021310  104412                          TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

                                ;++
                                ;       THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ;       TO NO LONGER BE TESTED.
                                ;--

                                        BGNDU
021312                          L$DU::
021312

                                        EXIT    DU
021312  000167                  .WORD   J$JMP
021314  000000                  .WORD   L10030-2-.

                                        ENDDU
021316                          L10030:
021316
021316  104453                  TRAP    C$DU
```

```
                                .SBTTL  ADD     UNIT SECTION

                                ;++
                                ;       THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
                                ;       TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
                                ;       TO THE TEST CYCLE.
                                ;--

021320                                  BGNAU
021320                          L$AU::

021320                                  EXIT    AU
021320  000167                          .WORD   J$JMP
021322  000000                          .WORD   L10031-2-.

021324                                  ENDAU
021324                          L10031:
021324  104452                          TRAP    C$AU
```

```
                                .SBTTL HARDWARE TESTS
1
2
```

```
                                .SBTTL  TEST 1: DISK EXERCISER

                                        BGNTST
021326                          T1::
021326
                        CMP     #1,TNUM                 ;CHECK IF TEST 1 WAS IN PROGRESS
021326  022737  000001  002202  BNE     T4STRT          ;BRANCH IF NOT
021334  001053                  CMP     #ICONT,IFLAGS   ;CHECK IF HERE BY CONTINUE COMMAND
021336  022737  000002  002200  BNE     T4STRT          ;BRANCH IF NOT
021344  001047                  CLR     IFLAGS          ;CLEAR FLAGS FOR NEXT TIME HERE
021346  005037  002200          MOV     LBUFS,R4        ;GET LOG BUFFER POINTER
021352  013704  002366          BEQ     LOGCHK          ;   IF ZERO, NONE EXISTS
021356  001423                  PNTF    LOGM1           ;INTRODUCE ERROR LOG
021360                          CLR     LBUFS           ;CLEAR START ADDRESS TO ERASE BUFFER
021370  005037  002366  LOGOUT: MOV     (R4)+,R5        ;GET CONTROLLER TABLE ADDRESS
021374  012405                  CALL    PNTERR          ;PRINT ERROR REPORT
021376  004737  014232          ADD     #<HC.BSZ-2>,R4  ;BUMP POINTER TO NEXT ENTRY
021402  062704  000104          CMP     R4,LBUFN        ;CHECK IF AT END
021406  020437  002370          BLO     LOGOUT          ;PRINT ALL ENTRIES
021412  103770                  PNTF    LOGM2
021414                          BR      T4CON
021424  000410
                        BIT     #SM.LOG,SFPTBL+SO.BIT   ;CHECK IF LOG ENABLED
021426  032737  001000  002144  LOGCHK: BEQ     T4CON
021434  001404                  PNTF    LOGM3           ;REPORT LOG EMPTY
021436                  TST     URNING          ;CHECK IF ANY CONTROLLERS STILL RUNNING
021446  005737  002206  T4CON:  BEQ     T4STRT          ;RESTART IF NOT
021452  001404                  CALL    RESPDM          ;CONTINUE BY RESPONDING TO REQUESTS
021454  004737  010374          JMP     T4WAIT          ;END OF TEST WHEN DONE
021460  000137  021756
                        ;
                        ;       START TEST
                        ;
                T4STRT: MOV     #1,R1                   ;INITIALIZE TEST PARAMETERS
021464  012701  000001          CALL    TINIT
021470  004737  010234          BIT     #ISTRT!IREST,IFLAGS     ;HERE FROM OPERATOR COMMAND?
021474  032737  000014  002200  BEQ     T4RUN           ;RUN WITH PREVIOUS PARAMETERS IF NEW PASS
021502  001506                  BIT     #SM.MAN,SFPTBL+SO.BIT   ;MANUAL INTERVENTION MODE?
021504  032737  000200  002144  BEQ     T4DEF           ;IF NOT, SET UP DEFAULT PARAMETERS
021512  001450                  MANUAL                  ;MANUAL INTERVENTION ALLOWED?
021514                          TRAP    C$MANI
021514  104450                  BNCOMPLETE T4DEFW       ;IF NOT, GIVE WARNING
021516                          BCC     T4DEFW
021516  103042
                        ;
                        ;       INPUT PARAMETERS
                        ;
                1$:     CLR     UCNT                    ;CLEAR COUNT OF UNITS USING PATTERN 16
021520  005037  002210          MOV     CTABS,R5        ;GET ADDRESS OF 1ST CONTROLLER TABLE
021524  013705  002170  T4PRM1: MOV     #4.,R2          ;GET COUNT OF DRIVE TABLES
021530  012702  000004          MOV     R5,R4           ;GET FIRST DRIVE TABLE POINTER
021534  010504                  ADD     #C.DRO,R4
021536  062704  000016  T4PRM2: MOV     (R4)+,R3        ;GET DRIVE TABLE ADDRESS
021542  012403                  BEQ     T4PRM4          ;GO TO NEXT CONTROLLER IF NONE
021544  001416                  BIT     #DT.AVL,D.UNIT(R3)      ;SEE IF TO BE TESTED
021546  032763  100000  000002  BNE     T4PRM3
021554  001010                  CALL    T4QUES          ; ASK CZUDIO QUESTIONS
021556  004737  022000          ;               INPUT - NONE
                        CMP     #16.,D.PAT(R3)
021562  022763  000020  000006
```

```
     59 021570  001002                          BNE     T4PRM3
     60 021572  005237  002210                  INC     UCNT
     61 021576  005302                T4PRM3:   DEC     R2                      ;COUNT DRIVE TABLES
     62 021600  001360                          BNE     T4PRM2                  ;GO LOOK AT NEXT
     63 021602  062705  000034        T4PRM4:   ADD     #C.SIZE,R5              ;GO TO NEXT CONTROLLER
     64 021606  005715                          TST     (R5)                    ;  IF THERE IS ONE
     65 021610  001347                          BNE     T4PRM1
     66                                     ;
     67                                     ;       NOW GET DATA PATTERN 16 IF SELECTED BY ANY DRIVE
     68                                     ;
     69 021612  012701  002256                  MOV     #PAT16C,R1              ; R1 -> PAT16C FOR INPUT
     70 021616  004737  024014                  CALL    T4QUE2                  ; ASK LAST OF THE CZUDIO QUESTIONS
     71                                     ;                                  ;       INPUT - R1 -> PATTERN 16 SIZE
     72 021622                                  ASSUME PAT16W EQ PAT16C+2
     73 021622  000436                          BR      T4RUN
     74                                     ;
     75                                     ;       GIVE WARNING MANUAL INTERVENTION NOT ALLOWED, IF NEEDED
     76                                     ;
     77 021624                         T4DEFW:  PNTF    T4WARN
     78                                     ;
     79                                     ;       SET UP DEFAULT PARAMETERS
     80                                     ;
     81 021634  013705  002170         T4DEF:   MOV     CTABS,R5                ;GET ADDRESS OF 1ST CONTROLLER TABLE
     82 021640  012702  000004         T4DEFA:  MOV     #4.,R2                  ;GET COUNT OF DRIVE TABLES
     83 021644  010504                          MOV     R5,R4                   ;GET FIRST DRIVE TABLE POINTER
     84 021646  062704  000016                  ADD     #C.DRO,R4
     85 021652  012403                T4DEFB:   MOV     (R4)+,R3                ;GET DRIVE TABLE ADDRESS
     86 021654  001415                          BEQ     T4DEFE                  ;GO TO NEXT CONTROLLER IF NONE
     87 021656  062703  000004                  ADD     #D.PRM,R3
     88 021662  042713  157777                  BIC     #^C<D.DCY>,(R3)         ;INITIALIZE ALL PARAMETER BITS
     89 021666  052723  011012                  BIS     #DDEF,(R3)+
     90 021672  012700  000067                  MOV     #55.,R0
     91 021676  005023                T4DEFC:   CLR     (R3)+
     92 021700  005300                          DEC     R0
     93 021702  001375                          BNE     T4DEFC
     94 021704  005302                T4DEFD:   DEC     R2                      ;COUNT DRIVE TABLES
     95 021706  001361                          BNE     T4DEFB                  ;GO LOOK AT NEXT
     96 021710  062705  000034        T4DEFE:   ADD     #C.SIZE,R5              ;GO TO NEXT CONTROLLER
     97 021714  005715                          TST     (R5)                    ;  IF THERE IS ONE
     98 021716  001350                          BNE     T4DEFA
     99                                     ;
    100                                     ;       START TEST
    101                                     ;
    102 021720  006137  002200         T4RUN:   ROL     IFLAGS                  ;CLEAR FLAGS FOR NEXT TIME HERE
    103 021724  042737  177757  002200         BIC     #^C<ISTRTH>,IFLAGS      ;HOLD START FOR DMRQ4 REQUEST
    104                                ;;       MOV     #1,R1                   ;INITIALIZE TEST PARAMETERS
    105                                ;;       CALL    TINIT
    106 021732  013737  002170  002174         MOV     CTABS,TSTTAB            ;GET ADDRESS OF 1ST CONTROLLER TABLE
    107 021740  013701  002172                 MOV     CTRLRS,R1               ;RUN DM PROGRAM ON ALL CONTROLLERS
    108 021744  004737  010300                 CALL    RUNDM
    109 021750  001402                         BEQ     T4WAIT
    110 021752  004737  010374                 CALL    RESPDM
    111 021756  032737  001000  002144 T4WAIT: BIT     #SM.LOG,SFPTBL+SO.BIT   ;CHECK IF LOG IS ENABLED
    112 021764  001402                         BEQ     T4EXIT                  ;EXIT IF NOT
    113                                                                        ;>>>>>>>>>>BREAK BACK TO MONITOR<<<<<<<<<<
    114 021766                                 BREAK
        021766  104422                         TRAP    C$BRK
```

```
    115 021770  000772                        BR      T4WAIT                  ;WAIT TILL STOPPED BY CONTROL C
    116 021772                          T4EXIT:                                 ;PRINT A STATISTICAL REPORT
    117 021772                                  DORPT
        021772  104424                          TRAP    C$DRPT
    118 021774                                  EXIT TST
        021774  104432                          TRAP    C$EXIT
        021776  003712                          .WORD   L10032-.
    119
```

```
                                .SBTTL  CZUDIO QUESTIONS

                                ;++
                                ;       ASK CZUDIO MANUAL INTERVENTION QUESTIONS
                                ;
                                ;       INPUTS:
                                ;               R5 - POINTER TO CONTROLLER TABLE
                                ;               R3 - POINTER TO DRIVE TABLE
                                ;               R2 AND R4 MUST BE PRESERVED
                                ;       OUTPUTS:
                                ;               DRIVE TABLE WITH NEW PARAMETERS
                                ;               R0 AND R1 CONTENTS DESTROYED
                                ;--

022000                          T4QUES: PUSH <R2,R4>
022004                                  PRINTF #T4QMED,D.UNIT(R3),(R5),(R3)     ;PRINT HEADER
022004  011346                         MOV     (R3),-(SP)
022006  011546                         MOV     (R5),-(SP)
022010  016346  000002                 MOV     D.UNIT(R3),-(SP)
022014  012746  025127                 MOV     #T4QMED,-(SP)
022020  012746  000004                 MOV     #4,-(SP)
022024  010600                         MOV     SP,R0
022026  104417                         TRAP    C$PNTF
022030  062706  000012                 ADD     #12,SP
022034  016337  000010  002214         MOV D.BB(R3),TEMP
022042                                  GMANID T4BB,TEMP,D,-1,0,16.,YES ;NUMBER OF BAD BLOCKS
022042  104443                         TRAP    C$GMAN
022044  000406                         BR      10000$
022046  002214                         .WORD   TEMP
022050  000052                         .WORD   T$CODE
022052  024122                         .WORD   T4BB
022054  177777                         .WORD   -1
022056  000000                         .WORD   T$LOLIM
022060  000020                         .WORD   T$HILIM
022062                          10000$:
022062  013763  002214  000010         MOV TEMP,D.BB(R3)
022070  001424                         BEQ T4Q02

022072  010304                         MOV R3,R4                       ;GET POINTER TO STORAGE
022074  062704  000012                 ADD #D.BB01,R4                  ;  FOR BAD BLOCKS
022100  013701  002214                 MOV TEMP,R1                     ;GET COUNT OF BLOCKS TO INPUT
022104  004737  013740         T4Q01:  CALL BLD28                      ;BUILD DEFAULT ANSWER
022110                                  GMANID T4BBI,TEMP,A,-1,0,9..YES ;BAD BLOCK
022110  104443                         TRAP    C$GMAN
022112  000406                         BR      10001$
022114  002214                         .WORD   TEMP
022116  000152                         .WORD   T$CODE
022120  024220                         .WORD   T4BBI
022122  177777                         .WORD   -1
022124  000000                         .WORD   T$LOLIM
022126  000011                         .WORD   T$HILIM
022130                          10001$:
022130  004737  014042                 CALL CNV28                      ;CONVERT TO BINARY
022134  103763                         BCS T4Q01                       ;REPEAT UNTIL RIGHT
022136  005301                         DEC R1                          ;DECREMENT COUNT
022140  001361                         BNE T4Q01                       ;GET ALL NUMBERS
022142                          T4Q02:
```

```
     32 022142  005037  025706                    CLR     ANYMOR                          ; DEFAULT ANYMORE WITH 0
     33 022146                                     GMANIL  T4DMN,ANYMOR,ANY.YS,YES
        022146  104443                             TRAP    C$GMAN
        022150  000404                             BR      10002$
        022152  025706                             .WORD   ANYMOR
        022154  000130                             .WORD   T$CODE
        022156  024147                             .WORD   T4DMN
        022160  000001                             .WORD   ANY.YS
        022162                          10002$:
     34 022162  032737  000001  025706             BIT     #ANY.YS,ANYMOR          ;DO WE ASK ANY MORE QUESTIONS?
     35 022170  001002                             BNE     1$                      ;  IF SO, CONTINUE
     36 022172  000137  024006                     JMP     T4Q30                   ; ELSE EXIT
     37 022176                          1$:
     38 022176  016337  000004  002214             MOV D.PRM(R3),TEMP              ; GET PARAMETER BITS
     39 022204                                     GMANIL  T4RO,TEMP,D.RO,YES      ; READ ONLY?
        022204  104443                             TRAP    C$GMAN
        022206  000404                             BR      10003$
        022210  002214                             .WORD   TEMP
        022212  000130                             .WORD   T$CODE
        022214  024232                             .WORD   T4RO
        022216  004000                             .WORD   D.RO
        022220                          10003$:
     40 022220  032737  004000  002214             BIT #D.RO,TEMP                  ; CHECK IF READ ONLY
     41 022226  001404                             BEQ T4Q03                       ; IF NOT, GO TO WRITE ONLY QUESTION
     42 022230  042737  002000  002214             BIC #D.WO,TEMP                  ; ELSE, CLEAR WRITE ONLY BIT
     43 022236  000432                             BR T4Q05                        ; AND BRANCH AROUND WRITE ONLY QUESTION
     44 022240                          T4Q03:
     45 022240                                     GMANIL  T4WO,TEMP,D.WO,YES      ; WRITE ONLY?
        022240  104443                             TRAP    C$GMAN
        022242  000404                             BR      10004$
        022244  002214                             .WORD   TEMP
        022246  000130                             .WORD   T$CODE
        022250  024244                             .WORD   T4WO
        022252  002000                             .WORD   D.WO
        022254                          10004$:
     46 022254                                     GMANIL  T4WCA,TEMP,D.WCA,YES    ; CHECK ALL WRITES
        022254  104443                             TRAP    C$GMAN
        022256  000404                             BR      10005$
        022260  002214                             .WORD   TEMP
        022262  000130                             .WORD   T$CODE
        022264  024257                             .WORD   T4WCA
        022266  000004                             .WORD   D.WCA
        022270                          10005$:
     47 022270  032737  000004  002214             BIT #D.WCA,TEMP                 ; CHECK ANSWER
     48 022276  001007                             BNE T4Q04                       ; BRANCH IF YES
     49 022300                                     GMANIL  T4WCR,TEMP,D.WC,YES     ; RANDOMLY CHECK WRITES
        022300  104443                             TRAP    C$GMAN
        022302  000404                             BR      10006$
        022304  002214                             .WORD   TEMP
        022306  000130                             .WORD   T$CODE
        022310  024313                             .WORD   T4WCR
        022312  000010                             .WORD   D.WC
        022314                          10006$:
     50 022314  000403                             BR T4Q05
     51 022316  052737  000010  002214  T4Q04:     BIS #D.WC,TEMP                  ; BOTH BITS GET SET
     52 022324  013763  002214  000004  T4Q05:     MOV TEMP,D.PRM(R3)              ; PUT PARAM BITS BACK
     53 022332  016337  000006  002214             MOV D.PAT(R3),TEMP
```

```
    54 022340                                    GMANID T4DP,TEMP,D,-1,0,16.,YES ; DATA PATTERN
       022340  104443                            TRAP    C$GMAN
       022342  000406                            BR      10007$
       022344  002214                            .WORD   TEMP
       022346  000052                            .WORD   T$CODE
       022350  024354                            .WORD   T4DP
       022352  177777                            .WORD   -1
       022354  000000                            .WORD   T$LOLIM
       022356  000020                            .WORD   T$HILIM
       022360                           10007$:
    55 022360  013763  002214  000006            MOV TEMP,D.PAT(R3)
    56 022366  001007                            BNE     1$                      ; IF NOT 0, BRANCH
    57 022370  022737  000020  002214            CMP     #16.,TEMP               ; IS PAT 16 SELETED?
    58 022376  001403                            BEQ     1$                      ; IF SO, BRANCH
    59 022400  042737  000001  025706            BIC     #ANY.YS,ANYMOR          ; CLEAR BIT
    60 022406  016337  000004  002214  1$:       MOV D.PRM(R3),TEMP              ;GET PARAM BITS AGAIN
    61 022414  032737  004000  002214  T4Q06:    BIT #D.RO,TEMP                  ;BYPASS NEXT 3 IF ONLY WRITING
    62 022422  001010                            BNE T4Q07
    63 022424  032737  002000  002214            BIT #D.WO,TEMP
    64 022432  001404                            BEQ T4Q07
    65 022434  032737  000010  002214            BIT #D.WC,TEMP
    66 022442  001432                            BEQ T4Q09
    67 022444                           T4Q07:   GMANIL T4ECC,TEMP,D.ECC,YES     ;ENABLE ECC
       022444  104443                            TRAP    C$GMAN
       022446  000404                            BR      10010$
       022450  002214                            .WORD   TEMP
       022452  000130                            .WORD   T$CODE
       022454  024422                            .WORD   T4ECC
       022456  010000                            .WORD   D.ECC
       022460                           10010$:
    68 022460                                    GMANIL T4DCA,TEMP,D.DCA,YES     ;COMPARE ALL DATA
       022460  104443                            TRAP    C$GMAN
       022462  000404                            BR      10011$
       022464  002214                            .WORD   TEMP
       022466  000130                            .WORD   T$CODE
       022470  024455                            .WORD   T4DCA
       022472  000001                            .WORD   D.DCA
       022474                           10011$:
    69 022474  032737  000001  002214            BIT #D.DCA,TEMP                 ;CHECK ANSWER
    70 022502  001007                            BNE T4Q08                       ;BRANCH IF YES
    71 022504                                    GMANIL T4DCR,TEMP,D.DC,YES      ;RANDOMLY CHECK WRITES
       022504  104443                            TRAP    C$GMAN
       022506  000404                            BR      10012$
       022510  002214                            .WORD   TEMP
       022512  000130                            .WORD   T$CODE
       022514  024503                            .WORD   T4DCR
       022516  000002                            .WORD   D.DC
       022520                           10012$:
    72 022520  000403                            BR T4Q09
    73 022522  052737  000002  002214  T4Q08:    BIS #D.DC,TEMP                  ;BOTH BITS GET SET
    74 022530                           T4Q09:   GMANIL T4RET,TEMP,D.RET,YES     ;ENABLE RETRIES
       022530  104443                            TRAP    C$GMAN
       022532  000404                            BR      10013$
       022534  002214                            .WORD   TEMP
       022536  000130                            .WORD   T$CODE
       022540  024536                            .WORD   T4RET
       022542  001000                            .WORD   D.RET
```

```
                022544                          10013$:
     75         022544  005137  002214                  COM TEMP
     76         022550                          GMANIL T4SEK,TEMP,D.SEQ,YES     ;ENABLE SEEKS
                022550  104443                  TRAP    C$GMAN
                022552  000404                  BR      10014$
                022554  002214                  .WORD   TEMP
                022556  000130                  .WORD   T$CODE
                022560  024555                  .WORD   T4SEK
                022562  000100                  .WORD   D.SEQ
                022564                          10014$:
     77         022564  005137  002214                  COM TEMP                        ;COMPLIMENTED
     78         022570  013763  002214  000004          MOV TEMP,D.PRM(R3)
     79
     80         022576  005037  002214                  CLR TEMP                        ;DETERMINE DEFAULT SELECTION
     81         022602  032763  000040  000004          BIT #D.BE,D.PRM(R3)             ;IF D.BE SET - LOAD 1
     82         022610  001403                  BEQ T4Q10                       ;IF D.CYL CLEAR - LOAD 0
     83         022612  005237  002214                  INC TEMP                        ;IF D.BEC CONTAINS 0 - LOAD 4
     84         022616  000422                  BR T4Q11                        ;IF D.TR SET - LOAD 2
     85         022620  032763  000400  000004  T4Q10:  BIT #D.CYL,D.PRM(R3)          ;LOAD 3
     86         022626  001416                  BEQ T4Q11
     87         022630  012737  000004  002214          MOV #4,TEMP
     88         022636  005763  000112                  TST D.BEC(R3)
     89         022642  001410                  BEQ T4Q11
     90         022644  005337  002214                  DEC TEMP
     91         022650  032763  000020  000004          BIT #D.TR,D.PRM(R3)
     92         022656  001402                  BEQ T4Q11
     93         022660  005337  002214                  DEC TEMP
     94         022664                          T4Q11:  PRINTF  #T4OPT1                 ; ENTER AREA DESIRED:
                022664  012746  025247                  MOV     #T4OPT1,-(SP)
                022670  012746  000001                  MOV     #1,-(SP)
                022674  010600                  MOV     SP,R0
                022676  104417                  TRAP    C$PNTF
                022700  062706  000004                  ADD     #4,SP
     95         022704                          PRINTF  #T4OPT2                 ; 0 - ENTIRE DISK AREA
                022704  012746  025306                  MOV     #T4OPT2,-(SP)
                022710  012746  000001                  MOV     #1,-(SP)
                022714  010600                  MOV     SP,R0
                022716  104417                  TRAP    C$PNTF
                022720  062706  000004                  ADD     #4,SP
     96         022724                          PRINTF  #T4OPT3                 ; 1 - BEGIN/END SETS
                022724  012746  025342                  MOV     #T4OPT3,-(SP)
                022730  012746  000001                  MOV     #1,-(SP)
                022734  010600                  MOV     SP,R0
                022736  104417                  TRAP    C$PNTF
                022740  062706  000004                  ADD     #4,SP
     97         022744                          PRINTF  #T4OPT4                 ; 2 - TRACKS & CYLINDERS
                022744  012746  025405                  MOV     #T4OPT4,-(SP)
                022750  012746  000001                  MOV     #1,-(SP)
                022754  010600                  MOV     SP,R0
                022756  104417                  TRAP    C$PNTF
                022760  062706  000004                  ADD     #4,SP
     98         022764                          PRINTF  #T4OPT5                 ; 3 - GROUPS & CYLINDERS
                022764  012746  025454                  MOV     #T4OPT5,-(SP)
                022770  012746  000001                  MOV     #1,-(SP)
                022774  010600                  MOV     SP,R0
                022776  104417                  TRAP    C$PNTF
                023000  062706  000004                  ADD     #4,SP
```

```
     99 023004                                    PRINTF  #T4OPT6                    ; 4 - CYLINDERS ONLY
        023004  012746  025523                    MOV     #T4OPT6,-(SP)
        023010  012746  000001                    MOV     #1.-(SP)
        023014  010600                            MOV     SP,R0
        023016  104417                            TRAP    C$PNTF
        023020  062706  000004                    ADD     #4,SP
    100 023024                                    GMANID  T4OPT7,TEMP,D,-1,0,4,YES;
        023024  104443                            TRAP    C$GMAN
        023026  000406                            BR      10015$
        023030  002214                            .WORD   TEMP
        023032  000052                            .WORD   T$CODE
        023034  002462                            .WORD   T4OPT7
        023036  177777                            .WORD   -1
        023040  000000                            .WORD   T$LOLIM
        023042  000004                            .WORD   T$HILIM
        023044                          10015$:
    101 023044  005337  002214                    DEC     TEMP                    ; CHECK ANSWER
    102 023050  002004                            BGE     T4Q12                   ; IF NOT 0, SKIP
    103 023052  042763  000440  000004            BIC     #D.BE+D.CYL,D.PRM(R3)   ;    ELSE, CLEAR D.BE AND D.CYL
    104 023060  000467                            BR      T4Q19                   ;    EXIT QUESTIONS
    105 023062  005337  002214          T4Q12:    DEC     TEMP                    ; CHECK ANSWER
    106 023066  002013                            BGE     T4Q13                   ; IF NOT 1, SKIP
    107 023070  032763  000040  000004            BIT     #D.BE,D.PRM(R3)         ;    CHECK D.BE
    108 023076  001060                            BNE     T4Q19                   ;    BRANCH IF D.BE SET
    109 023100  052763  000040  000004            BIS     #D.BE,D.PRM(R3)         ;    SET B/E SET FLAG
    110 023106  042763  000400  000004            BIC     #D.CYL,D.PRM(R3)        ;    CLEAR CYLINDER FLAG
    111 023114  000436                            BR      T4Q16                   ;    GET B/E SETS
    112 023116  042763  000040  000004  T4Q13:    BIC     #D.BE,D.PRM(R3)         ; CLEAR B/E SET FLAG
    113 023124  022737  000002  002214            CMP     #2,TEMP                 ;
    114 023132  001006                            BNE     T4Q14                   ; IF NOT 4, SKIP
    115 023134  052763  000400  000004            BIS     #D.CYL,D.PRM(R3)        ;    SET CYLINDER FLAG
    116 023142  005063  000112                    CLR     D.BEC(R3)               ;    CLEAR B/E SET COUNT
    117 023146  000434                            BR      T4Q19                   ;    GET CYLINDERS
    118 023150                          T4Q14:    PUSH    D.PRM(R3)               ; IF 2 OR 3
    119 023154  052763  000420  000004            BIS     #D.CYL+D.TR,D.PRM(R3)   ;    SAVE D.PRM BITS
    120 023162  005337  002214                    DEC     TEMP                    ;    SET D.CYL AND D.TR
    121 023166  001403                            BEQ     T4Q15                   ; IF 3
    122 023170  042763  000020  000004            BIC     #D.TR,D.PRM(R3)         ;    CLEAR TRACK FLAG
    123 023176  022663  000004          T4Q15:    CMP     (SP)+,D.PRM(R3)         ;    CHECK OLD PARAMETER BITS
    124 023202  001003                            BNE     T4Q16                   ;    IF D.CYL OR D.TR CHANGED, SKIP
    125 023204  005763  000112                    TST     D.BEC(R3)               ;    CHECK B/E SET COUNT
    126 023210  001013                            BNE     T4Q19                   ;    IF NOT 0, GET CYLINDERS
    127 023212  012763  000001  000112  T4Q16:    MOV     #1,D.BEC(R3)            ; LOAD 1 INTO B/E SET COUNT
    128 023220  010304                  T4Q17:    MOV     R3,R4                   ; GET POINTER TO DRIVE TABLE
    129 023222  062704  000114                    ADD     #D.BGN1,R4              ; POINT TO 1ST B/E SET
    130 023226  012701  000020                    MOV     #16.,R1                 ; GET COUNT OF WORDS TO CLEAR
    131 023232  005024                  T4Q18:    CLR     (R4)+                   ; CLEAR A WORD
    132 023234  005301                            DEC     R1                      ; DECREMENT WORD COUNTER
    133 023236  001375                            BNE     T4Q18                   ; IF NOT DONE, LOOP
    134 023240  032763  000040  000004  T4Q19:    BIT     #D.BE,D.PRM(R3)         ; CHECK IF WE'RE USING B/E SETS
    135 023246  001460                            BEQ     T4Q22                   ; IF D.BE CLEAR, USE CYLINDERS
    136 023250  016337  000112  002214            MOV     D.BEC(R3),TEMP          ; GET PREVIOUS B/E SET COUNT
    137 023256                                    GMANID  T4BE,TEMP,D,-1,1,4,YES  ; NUMBER OF B/E SETS?
        023256  104443                            TRAP    C$GMAN
        023260  000406                            BR      10016$
        023262  002214                            .WORD   TEMP
        023264  000052                            .WORD   T$CODE
```

```
        023266  024600                          .WORD   T4BE
        023270  177777                          .WORD   -1
        023272  000001                          .WORD   T$LOLIM
        023274  000004                          .WORD   T$HILIM
        023276                          10016$:
138     023276  013763  002214  000112          MOV     TEMP,D.BEC(R3)          ; SAVE NEW B/E SET COUNT
139     023304  013701  002214                  MOV     TEMP,R1                 ; INIT LOOP COUNTER
140     023310  010304                          MOV     R3,R4                   ; GET POINTER TO STORAGE AREA
141     023312  062704  000114                  ADD     #D.BGN1,R4              ; POINT TO START OF 1ST B/E SET
142     023316  004737  013740          T4Q20:  CALL    BLD28                   ; BUILD DEFAULT VALUE
143     023322                                  GMANID  T4BEG,TEMP,A,-1,0,9.,YES; BEGIN BLOCK?
        023322  104443                          TRAP    C$GMAN
        023324  000406                          BR      10017$
        023326  002214                          .WORD   TEMP
        023330  000152                          .WORD   T$CODE
        023332  024631                          .WORD   T4BEG
        023334  177777                          .WORD   -1
        023336  000000                          .WORD   T$LOLIM
        023340  000011                          .WORD   T$HILIM
        023342                          10017$:
144     023342  004737  014042                  CALL    CNV28                   ; CHECK RESPONSE & SAVE IN TABLE
145     023346  103763                          BCS     T4Q20                   ; BRANCH ON ERROR
146     023350  004737  013740          T4Q21:  CALL    BLD28                   ; BUILD DEFAULT VALUE
147     023354                                  GMANID  T4END,TEMP,A,-1,0,9.,YES; END BLOCK?
        023354  104443                          TRAP    C$GMAN
        023356  000406                          BR      10020$
        023360  002214                          .WORD   TEMP
        023362  000152                          .WORD   T$CODE
        023364  024645                          .WORD   T4END
        023366  177777                          .WORD   -1
        023370  000000                          .WORD   T$LOLIM
        023372  000011                          .WORD   T$HILIM
        023374                          10020$:
148     023374  004737  014042                  CALL    CNV28                   ; CHECK RESPONSE & SAVE IN TABLE
149     023400  103763                          BCS     T4Q21                   ; BRANCH ON ERROR
150     023402  005301                          DEC     R1                      ; DECREMENT LOOP COUNT
151     023404  001344                          BNE     T4Q20                   ; IF NOT DONE, LOOP
152     023406  000577                          BR      T4Q30                   ;     ELSE, EXIT QUESTIONS
153     023410  032763  000400  000004  T4Q22:  BIT #D.CYL,D.PRM(R3)            ;IF D.CYL CLEAR - ALL DONE
154     023416  001573                          BEQ T4Q30
155     023420  005763  000112                  TST D.BEC(R3)                   ;IF D.BEC CLEAR - GO RIGHT TO B/E CYLS
156     023424  001526                          BEQ T4Q27
157     023426  010304                          MOV R3,R4
158     023430  062704  000112                  ADD #D.BEC,R4
159     023434  032763  000020  000004          BIT #D.TR,D.PRM(R3)             ;LOOK AT D.TR.TO DETERMINE QUESTION
160     023442  001434                          BEQ T4Q24
161     023444  011437  002214                  MOV (R4),TEMP
162     023450                                  GMANID T4TRC,TEMP,D,-1,1,7,YES  ;NUMBER OF TRACKS
        023450  104443                          TRAP    C$GMAN
        023452  000406                          BR      10021$
        023454  002214                          .WORD   TEMP
        023456  000052                          .WORD   T$CODE
        023460  024657                          .WORD   T4TRC
        023462  177777                          .WORD   -1
        023464  000001                          .WORD   T$LOLIM
        023466  000007                          .WORD   T$HILIM
        023470                          10021$:
```

```
    163 023470  013714  002214                  MOV TEMP,(R4)
    164 023474  012401                          MOV (R4)+,R1                    ;GET COUNT OF TRACKS
    165 023476  011437  002214          T4Q23:  MOV (R4),TEMP
    166 023502                                  GMANID T4TRAK,TEMP,D,-1,0,255.,YES      ;TRACK
        023502  104443                          TRAP    C$GMAN
        023504  000406                          BR      10022$
        023506  002214                          .WORD   TEMP
        023510  000052                          .WORD   T$CODE
        023512  024710                          .WORD   T4TRAK
        023514  177777                          .WORD   -1
        023516  000000                          .WORD   T$LOLIM
        023520  000377                          .WORD   T$HILIM
        023522                          10022$:
    167 023522  013724  002214                  MOV TEMP,(R4)+
    168 023526  005301                          DEC R1
    169 023530  001362                          BNE T4Q23
    170 023532  000433                          BR T4Q26
    171 023534  011437  002214          T4Q26:  MOV (R4),TEMP
    172 023540                                  GMANID T4GRC,TEMP,D,-1,1,7,YES  ;NUMBER OF GROUPS
        023540  104443                          TRAP    C$GMAN
        023542  000406                          BR      10023$
        023544  002214                          .WORD   TEMP
        023546  000052                          .WORD   T$CODE
        023550  024716                          .WORD   T4GRC
        023552  177777                          .WORD   -1
        023554  000001                          .WORD   T$LOLIM
        023556  000007                          .WORD   T$HILIM
        023560                          10023$:
    173 023560  013714  002214                  MOV TEMP,(R4)
    174 023564  012401                          MOV (R4)+,R1                    ;GET COUNT OF GROUPS
    175 023566  011437  002214          T4Q25:  MOV (R4),TEMP
    176 023572                                  GMANID T4GRP,TEMP,D,-1,0,255.,YES       ;GROUP
        023572  104443                          TRAP    C$GMAN
        023574  000406                          BR      10024$
        023576  002214                          .WORD   TEMP
        023600  000052                          .WORD   T$CODE
        023602  024747                          .WORD   T4GRP
        023604  177777                          .WORD   -1
        023606  000000                          .WORD   T$LOLIM
        023610  000377                          .WORD   T$HILIM
        023612                          10024$:
    177 023612  013724  002214                  MOV TEMP,(R4)+
    178 023616  005301                          DEC R1
    179 023620  001362                          BNE T4Q25
    180 023622  016337  000162  002214  T4Q26:  MOV D.ECYL+2(R3),TEMP
    181 023630  005137  002214                  COM TEMP
    182 023634                                  GMANIL T4CYL,TEMP,BIT15,YES     ;WISH TO LIMIT CYLINDERS
        023634  104443                          TRAP    C$GMAN
        023636  000404                          BR      10025$
        023640  002214                          .WORD   TEMP
        023642  000130                          .WORD   T$CODE
        023644  024755                          .WORD   T4CYL
        023646  100000                          .WORD   BIT15
        023650                          10025$:
    183 023650  005737  002214                  TST TEMP
    184 023654  100412                          BMI T4Q27
    185 023656  005063  000154                  CLR D.BCYL(R3)
```

```
                                                CLR D.BCYL+2(R3)
                                                CLR D.ECYL(R3)
                                                MOV #-1,D.ECYL+2(R3)
                                                BR T4Q30
                                        T4Q27:  TST D.ECYL+2(R3)
                                                BGE T4Q27A
                                                CLR D.ECYL+2(R3)
                                        T4Q27A: MOV R3,R4
                                                ADD #D.BCYL,R4
                                        T4Q28:  CALL BLD28
                                                GMANID T4CYLB,TEMP,A,-1,0,9.,YES        ;STARTING CYLINDER
023726  104443                                  TRAP    C$GMAN
023730  000406                                  BR      10026$
023732  002214                                  .WORD   TEMP
023734  000152                                  .WORD   T$CODE
023736  025010                                  .WORD   T4CYLB
023740  177777                                  .WORD   -1
023742  000000                                  .WORD   T$LOLIM
023744  000011                                  .WORD   T$HILIM
023746                                  10026$:
023746  004737  014042                          CALL CNV28
023752  103763                                  BCS T4Q28
023754  004737  013740                T4Q29:  CALL BLD28
                                                GMANID T4CYLE,TEMP,A,-1,0,9.,YES        ;ENDING CYLINDER
023760  104443                                  TRAP    C$GMAN
023762  000406                                  BR      10027$
023764  002214                                  .WORD   TEMP
023766  000152                                  .WORD   T$CODE
023770  025032                                  .WORD   T4CYLE
023772  177777                                  .WORD   -1
023774  000000                                  .WORD   T$LOLIM
023776  000011                                  .WORD   T$HILIM
024000                                  10027$:
024000  004737  014042                          CALL CNV28
024004  103763                                  BCS T4Q29
024006                                  T4Q30:  POP <R4,R2>
024012  000207                                  RETURN
                                        ;NOW GET DATA PATTERN 16 IF SELECTED BY ANY DRIVE
                                        ;
                                        ;       INPUT R1 POINTS TO PATTERN 16 SIZE

024014                                  T4QUE2:
024014  032737  000001  025706                  BIT     #ANY.YS,ANYMOR          ; DO WE ASK THESE QUESTIONS?
024022  001436                                  BEQ     T4QU2E                  ; IF NOT, EXIT
024024  011137  002214                          MOV     (R1),TEMP
024030                                          GMANID T4DPC,TEMP,D,-1,0,16.,YES; COUNT OF WORDS
024030  104443                                  TRAP    C$GMAN
024032  000406                                  BR      10030$
024034  002214                                  .WORD   TEMP
024036  000052                                  .WORD   T$CODE
024040  025052                                  .WORD   T4DPC
024042  177777                                  .WORD   -1
024044  000000                                  .WORD   T$LOLIM
024046  000020                                  .WORD   T$HILIM
024050                                  10030$:
024050  013721  002214                          MOV     TEMP,(R1)+              ; GET COUNT OF WORDS
024054  010104                                  MOV     R1,R4                   ; R4 HAS ADDRESS OF STORAGE
```

```
    216 024056  013701  002214                  MOV     TEMP,R1
    217 024062  001416                          BEQ     T4QU2E                  ; IF 0, NO DATA WORDS
    218 024064  011437  002214          T4PRM5: MOV (R4),TEMP
    219 024070                                  GMANID T4DPD,TEMP,0,-1,0,-1,YES ;DATA WORD
        024070  104443                          TRAP    C$GMAN
        024072  000406                          BR      10031$
        024074  002214                          .WORD   TEMP
        024076  000032                          .WORD   T$CODE
        024100  025115                          .WORD   T4DPD
        024102  177777                          .WORD   -1
        024104  000000                          .WORD   T$LOLIM
        024106  177777                          .WORD   T$HILIM
        024110                          10031$:
    220 024110  013724  002214                  MOV TEMP,(R4)+
    221 024114  005301                          DEC R1                          ;COUNT THE WORDS
    222 024116  001362                          BNE T4PRM5
    223 024120  000207                  T4QU2E: RETURN
    224
```

```
                                ;
                                ; UNFORMATTED QUESTIONS
                                ;

024122  116  125  115  T4BB:   .ASCIZ\NUMBER OF BAD BLOCKS\
024147  103  110  101  T4DMN:  .ASCIZ\CHANGE TESTING PARAMETERS FOR THIS DRIVE\
024220  102  101  104  T4BBI:  .ASCIZ\BAD BLOCK\
024232  122  105  101  T4RO:   .ASCIZ\READ ONLY\
024244  127  122  111  T4WO:   .ASCIZ\WRITE ONLY\
024257  103  110  105  T4WCA:  .ASCIZ\CHECK ALL WRITES BY READING\
024313  122  101  116  T4WCR:  .ASCIZ\RANDOMLY CHECK WRITES BY READING\
024354  104  101  124  T4DP:   .ASCIZ\DATA PATTERN - 0 FOR RANDOM SELECTION\
024422  105  116  101  T4ECC:  .ASCIZ\ENABLE ECC DATA CORRECTION\
024455  103  117  115  T4DCA:  .ASCIZ\COMPARE ALL DATA READ\
024503  122  101  116  T4DCR:  .ASCIZ\RANDOMLY COMPARE DATA READ\
024536  105  116  101  T4RET:  .ASCIZ\ENABLE RETRIES\
024555  122  101  116  T4SEK:  .ASCIZ\RANDOM ACCESS MODE\
024600  116  125  115  T4BE:   .ASCIZ\NUMBER OF BEGIN/END SETS\
024631  102  105  107  T4BEG:  .ASCIZ\BEGIN BLOCK\
024645  105  116  104  T4END:  .ASCIZ\END BLOCK\
024657  116  125  115  T4TRC:  .ASCIZ\NUMBER OF TRACKS TO TEST\
024710  124  122  101  T4TRAK: .ASCIZ\TRACK\
024716  116  125  115  T4GRC:  .ASCIZ\NUMBER OF GROUPS TO TEST\
024747  107  122  117  T4GRP:  .ASCIZ\GROUP\
024755  114  111  115  T4CYL:  .ASCIZ\LIMIT THE CYLINDERS TESTED\
025010  123  124  101  T4CYLB: .ASCIZ\STARTING CYLINDER\
025032  105  116  104  T4CYLE: .ASCIZ\ENDING CYLINDER\
025052  116  125  115  T4DPC:  .ASCIZ\NUMBER OF WORDS IN DATA PATTERN 16\
025115  104  101  124  T4DPD:  .ASCIZ\DATA WORD\

                                ; FORMATTED QUESTIONS

025127  045  116  045  T4QHED: .ASCIZ\%N%ATHE FOLLOWING QUESTIONS REFER TO UNIT %D2%A CONTROLLER AT %O6%A DRIVE %D3
%
025247  045  116  045  T4QPT1: .ASCIZ\%N%AENTER TEST AREA DESIRED:%N\
025306  045  101  040  T4QPT2: .ASCIZ\%A   0 - ENTIRE DISK AREA%N\
025342  045  101  040  T4QPT3: .ASCIZ\%A   1 - SPECIFIC BEGIN/END SETS%N\
025405  045  101  040  T4QPT4: .ASCIZ\%A   2 - SPECIFIC TRACKS & CYLINDERS%N\
025454  045  101  040  T4QPT5: .ASCIZ\%A   3 - SPECIFIC GROUPS & CYLINDERS%N\
025523  045  101  040  T4QPT6: .ASCIZ\%A   4 - SPECIFIC CYLINDERS\
025557  045  101  114  INP28A: .ASCIZ\%ALIMITS - LO= 0, HI= 268435455%N\
025621  045  101  111  INP28B: .ASCIZ\%AINVALID CHAR, TYPE DECIMAL NUMBER 0 TO 268435455%N\
                                .EVEN

025706  000000                  ANYMOR: .WORD   0       ; ANY MORE QUESTIONS
        000001                  ANY.YS = BIT0

                                ENDTST
025710                          L10032:
025710  104401                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ;       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.
                                ;--
                                        BGNHRD
025712  000014                          .WORD L10033-L$HARD/2
025714                          L$HARD::

                                        TABLE                                   ;START A TABLE DEFINITION
                                ITEM H.UBA      2       <CSR ADDRESS>
                                ITEM H.DRV      2       <DRIVE NUMBER>
                                ITEM H.PRM      2       <PROGRAM PARAMETERS>

        020000                  MM.CYL  == BIT13                        ; TEST CUSTOMER DATA AREA
                                END
                                                                        ; CSR ADDRESS OF CONTROLLER?
                                        GPRMA   MSGUBA,H.UBA,0,160000,177774,YES
025714  000031                          .WORD   T$CODE
025716  025744                          .WORD   MSGUBA
025720  160000                          .WORD   T$LOLIM
025722  177774                          .WORD   T$HILIM
                                                                        ; DRIVE #?
025724                                  GPRMD   MSGLDR,H.DRV,D,-1,0..255..YES
025724  001052                          .WORD   T$CODE
025726  025776                          .WORD   MSGLDR
025730  177777                          .WORD   -1
025732  000000                          .WORD   T$LOLIM
025734  000377                          .WORD   T$HILIM
                                                                        ; EXERCISE ON CUSTOMER DATA AREA?
025736                                  GPRML   MSGCST,H.PRM,MM.CYL,YES
025736  002130                          .WORD   T$CODE
025740  026006                          .WORD   MSGCST
025742  020000                          .WORD   MM.CYL

025744                                  ENDHRD
                                        .EVEN
025744                          L10033:

025744  103     123     122     MSGUBA: .ASCIZ  \CSR ADDRESS OF CONTROLLER\
025776  104     122     111     MSGLDR: .ASCIZ  \DRIVE #\
026006  105     130     105     MSGCST: .ASCIZ  \EXERCISE ON CUSTOMER DATA AREA\

                                        .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.
                                ;--

026046                                  BGNSFT
026046  000026                          .WORD L10034-L$SOFT/2
026050                          L$SOFT::
026050                          TABLE                                   ; START A TABLE DEFINITION
026050                          ITEM SO.EL      2       <ERROR LIMIT>
026050                          ITEM SO.XL      2       <DATA TRANSFER LIMIT (MEGABITS)>
026050                          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
026050                          END

                                                                        ; ENTER MANUAL INTERVENTION MODE
                                                                        ;       FOR SPECIAL DIAGNOSIS?'
026050                                  GPRML S.MAN,SO.BIT,SM.MAN,YES
026050  002130                          .WORD   T$CODE
026052  026124                          .WORD   S.MAN
026054  000200                          .WORD   SM.MAN
                                                                        ; ERROR LIMIT?
026056                                  GPRMD S.EL,SO.EL,D,-1,1.,-1.,YES
026056  000052                          .WORD   T$CODE
026060  026211                          .WORD   S.EL
026062  177777                          .WORD   -1
026064  000001                          .WORD   T$LOLIM
026066  177777                          .WORD   T$HILIM
                                                                        ; READ TRANSFER LIMIT IN MEGABYTES
                                                                        ;       - 0 FOR NO LIMIT?
026070                                  GPRMD S.XL,SO.XL,D,-1,0.,-1.,YES
026070  001052                          .WORD   T$CODE
026072  026225                          .WORD   S.XL
026074  177777                          .WORD   -1
026076  000000                          .WORD   T$LOLIM
026100  177777                          .WORD   T$HILIM
                                                                        ; SUPPRESS PRINTING SOFT ERRORS?
026102                                  GPRML S.SSF,SO.BIT,SM.SSF,YES
026102  002130                          .WORD   T$CODE
026104  026307                          .WORD   S.SSF
026106  000400                          .WORD   SM.SSF
                                                                        ;PRINT 'DO INITIAL WRITE ON START?'
026110                                  GPRML S.IW,SO.BIT,SM.IW,YES
026110  002130                          .WORD   T$CODE
026112  026345                          .WORD   S.IW
026114  040000                          .WORD   SM.IW
                                                                        ;PRINT 'ENABLE ERROR LOG?'
```

```
                                                GPRML S.LOG,SO.BIT,SM.LOG,YES
                                                .WORD   T$CODE
                                                .WORD   S.LOG
                                                .WORD   SM.LOG
                                                ENDSFT
                                                .EVEN
                                        L10034:
                                        S.MAN:  .ASCIZ  \ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS\
                                        S.EL:   .ASCIZ  \ERROR LIMIT\
                                        S.XL:   .ASCIZ  \READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT\
                                        S.SSF:  .ASCIZ  \SUPPRESS PRINTING SOFT ERRORS\
                                        S.IW:   .ASCIZ  \DO INITIAL WRITE ON START\
                                        S.LOG:  .ASCIZ  \ENABLE ERROR LOG\
                                                .EVEN
```

```
                                ;**
                                ;       THIS IS WHERE THE DM PROGRAMS WILL BE LINKED
                                ;--

                                        .DSABL AMA
000000                                  .PSECT DM,ABS                   ; DM PROGRAMS ARE LINKED HERE
000000                                  .PSECT END

                                ;**
                                ;       THIS IS A PATCH AREA THAT SHOULD BE INCLUDED IN ALL DIAGNOSTICS.
                                ;       THE SIZE IS ADJUSTED AS NEEDED.
                                ;--

000000                          $PATCH::
000000                                  .BLKW   16.

000040                                  LASTAD
                                        .EVEN
000040  000056'                         .WORD T$FREE
000042  000005                         .WORD T$SIZE
000044                          L$LAST::
```

```
 1
 2                                      ;**
 3                                      ;       HARDCODED P-TABLES MAY BE PLACED HERE BY USING THE SETUP MACROS.
 4                                      ;       THIS SECTION IS OPTIONAL AND SHOULD BE REMOVED IF IT IS NOT BEING
 5                                      ;       USED.  CHANGE THE POINTER MACRO ARGUMENT TO REFLECT THE REMOVAL.
 6                                      ;
 7                                      ;       THE P-TABLES ARE DELIMITED BY THE "BGNSETUP" AND "ENDSETUP" MACROS.
 8                                      ;       THE "BGNSETUP" MACRO HAS ONE ARGUMENT WHICH IS THE NUMBER OF
 9                                      ;       P-TABLE ENTRIES.   EACH ENTRY IS DELIMITED BY THE "BGNPTAB" AND
10                                      ;       "ENDPTAB" MACROS.  NEITHER OF THESE MACROS REQUIRE AN ARGUMENT.
11                                      ;--
12
13 000044                               BGNSETUP        1
14 000044                               BGNPTAB
   000044  000000                       .WORD   0
   000046  000003                       .WORD   L10037-./2-1
   000050                       L10035:
15 000050  172150                       .WORD   172150                  ; CSR ADDRESS
20 000052  000000                       .WORD   0.                      ; DRIVE NUMBER
22 000054  000000                       .WORD   0.                      ; COSTUMER DATA AREA
24 000056                               ENDPTAB
   000056                       L10037:
25 000056                               ENDSETUP
26          000001              .END
```

Symbol table

```
ADR   = 000020 G      CF.MSC= 000100        C$ERHR= 000056        DIV10   013672        D.CYL  = 000400
ALOCM   007332        CF.OTH= 000040        C$ERRO= 000060        DMFRST= 001000        D.DC   = 000002
ANYMOR  025706        CF.SMD= 000002        C$ERSF= 000054        DMMAIN= 000040        D.DCA  = 000001
ANY.YS= 000001        CF.TMS= 000020        C$ERSO= 000057        DMOVRL= 000004        D.DCY  = 020000
ARG.CT= 000000        CF.576= 000001        C$ESCA= 000010        DMPROG  002176        D.DRV  = 000000
ASS   = 000004        CLRBFL  014760        C$ESEG= 000005        DMRQA   012116        D.ECC  = 010000
ASSEMB= 000010        CLRBUF  014732        C$ESUB= 000003        DMRQB   012176        D.ECCC= 000176
BAS     003654        CNTINT  015272        C$ETST= 000001        DMRQC   012402        D.ECYL= 000160
BASLN   003655        CNTIST  015502        C$EXIT= 000032        DMRQD   012540        D.END1= 000120
BASL1   003573        CNTRSD  002364        C$GETB= 000026        DMRQE   012652        D.END2= 000130
BASL2   003611        CNTRSP  015776        C$GETW= 000027        DMRQ0   011164        D.END3= 000140
BASL3   003637        CNTSRV  015156 G      C$GMAN= 000043        DMRQ3   011304        D.END4= 000150
BASN0   003533        CNV28   014042        C$GPHR= 000042        DMRQ4   011326        D.MDAS= 000210
BASN4   003552        CON.A   007766        C$GPLO= 000030        DMRQ4X  011620        D.MERR= 000170
BELL  = 000007 G      CON.A1  007772        C$GPRI= 000040        DMRQ5   011622        D.IW   = 040000
BIT0  = 000001 G      CON.A2  010014        C$INIT= 000011        DMRQ5E  011650        D.PAT  = 000006
BIT00 = 000001 G      CON.D   010016        C$INLP= 000020        DMRQ6   011652        D.PRM  = 000004
BIT01 = 000002 G      CON.H   010030        C$MANI= 000050        DMRQ6E  011700        D.RET  = 001000
BIT02 = 000004 G      CON.N   010054        C$MEM = 000031        DMRQ7   011702        D.RO   = 004000
BIT03 = 000010 G      CON.N1  010060        C$MSG = 000023        DMRQ8   011734        D.SEEK= 000174
BIT04 = 000020 G      CON.O   010042        C$OPEN= 000034        DMRQ9   011754        D.SEQ  = 000100
BIT05 = 000040 G      CON.QU  007746        C$PNTB= 000014        DMTRLN= 000000        D.SERN= 000200
BIT06 = 000100 G      CON.QX  007764        C$PNTF= 000017        DSPSIZ= 000017        D.SERR= 000172
BIT07 = 000200 G      CON.R   010076        C$PNTS= 000016        DTABS   002166 G      D.SIZE= 000220
BIT08 = 000400 G      CON.R1  010120        C$PNTX= 000015        DT.AVL= 100000        D.SKER= 000206
BIT09 = 001000 G      CON.S   010134        C$QIO = 000377        DT.UNT= 000077        D.TR   = 000020
BIT1  = 000002 G      CON.S1  010140        C$RDBU= 000007        DUP   = 001000        D.UNIT= 000002
BIT10 = 002000 G      CR    = 000015 G      C$REFG= 000047        DU.DFL= 020000        D.WC   = 000010
BIT11 = 004000 G      CRLF    002544        C$RESE= 000033        DU.FTL= 050000        D.WCA  = 000004
BIT12 = 010000 G      CTABS   002170 G      C$REVI= 000003        DU.INF= 030000        D.WD   = 002000
BIT13 = 020000 G      CTRLRS  002172        C$RFLA= 000021        DU.QUE= 010000        D.XFRR= 000166
BIT14 = 040000 G      CT.AVL= 100000        C$RPT = 000025        DU.SPC= 060000        D.XFRW= 000164
BIT15 = 100000 G      CT.BRL= 007000        C$SEFG= 000046        DU.TER= 040000        D.ZERO= 140200
BIT2  = 000004 G      CT.CMD= 0000C4        C$SPRI= 000041        D.BB  = 000010        EF.BBR= 000200
BIT3  = 000010 G      CT.MSG= 000010        C$SVEC= 000037        D.BB01= 000012        EF.BBU= 000100
BIT4  = 000020 G      CT.REQ= 000020        C$TPRI= 000013        D.BB02= 000016        EF.CON= 000036 G
BIT5  = 000040 G      CT.RN = 000002        C.DR0 = 000016        D.BB03= 000022        EF.LOG= 000040
BIT6  = 000100 G      CT.UNT= 000077        C.DR1 = 000020        D.BB04= 000026        EF.NEW= 000035 G
BIT7  = 000200 G      CT.VEC= 000777        C.DR2 = 000022        D.BB05= 000032        EF.PWR= 000034 G
BIT8  = 000400 G      CT.VER= 000100        C.DR3 = 000024        D.BB06= 000036        EF.RES= 000037 G
BIT9  = 001000 G      C$AU  = 000052        C.FLG = 000012        D.BB07= 000042        EF.SEX= 000020
BLDCMD  014572        C$AUTO= 000061        C.HCOM= 000014        D.BB08= 000046        EF.STA= 000040 G
BLDC0   014640        C$BRK = 000022        C.JAD = 000010        D.BB09= 000052        ERRBLK  002154 G
BLDC1   014646        C$BSEG= 000004        C.JSR = 000006        D.BB10= 000056        ERRC    010202
BLD28   013740        C$BSUB= 000002        C.REF = 000032        D.BB11= 000062        ERRD    010214
BOE   = 000400 G      C$CEFG= 000045        C.SIZE= 000034        D.BB12= 000066        ERRLIM  002747
BRSAV   002254        C$CLCK= 000062        C.TO  = 000026        D.BB13= 000072        ERRME1  002603
CALR1   012656        C$CLEA= 000012        C.TOM = 000030        D.BB14= 000076        ERRMSG  002152 G
CALR2   012704        C$CLOS= 000035        C.UADR= 000000        D.BB15= 000102        ERRMSL  012274
CALR3   013002        C$CLP1= 000006        C.UNIT= 000002        D.BB16= 000106        ERRMSX  012376
CALR4   013016        C$CVEC= 000036        C.VEC = 000004        D.BCYL= 000154        ERRNBR  002150 G
CALR5   013072        C$DCLN= 000044        DDEF  = 011012        D.BE  = 000040        ERRTYP  002146 G
CALR6   013150        C$DODU= 000051        DEBUG = 000000        D.BEC = 000112        ERRVEC= 000004 G
CALR7   013164        C$DRPT= 000024        DFPTBL  002130 G      D.BGN1= 000114        ERR.SZ= 000011
CALR8   013202        C$DU  = 000053        DIAG  = 177777        D.BGN2= 000124        ERR.TB  010156
CALR9   013220        C$EDIT= 000003        DIAGMC= 000000        D.BGN3= 000134        ERR.TN  007160 G
CF.ATN= 000200        C$ERDF= 000055        DIVIDE  013634        D.BGN4= 000144        ERR002  006540 G
```

Symbol table

```
ERR003  006556 G        G$PRMD= 000002          I$AU  = 000041          L$CLEA  021304 G        L10011  006700
ERR004  006574 G        G$PRML= 000000          I$AUTO= 000041          L$CO    002032 G        L10012  006706
ERR006  006606 G        G$RADA= 000140          I$CLN = 000041          L$DEPO  002011 G        L10013  006724
ERR014  006620 G        G$RADB= 000000          I$DU  = 000041          L$DESC  002420 G        L10014  006736
ERR030  006636 G        G$RADD= 000040          I$HRD = 000041          L$DESP  002076 G        L10015  006750
ERR031  006652 G        G$RADL= 000120          I$INIT= 000041          L$DEVP  002060 G        L10016  006764
ERR032  006664 G        G$RADO= 000020          I$MOD = 000041          L$DISP  002124 G        L10017  007316
ERR033  006702 G        G$XFER= 000004          I$MSG = 000041          L$DLY   002116 G        L10020  015154
ERR035  006710 G        G$YES = 000010          I$PROT= 000040          L$DTP   002040 G        L10021  015164
ERR036  006726 G        HCOMM   007356          I$PTAB= 000041          L$DTYP  002034 G        L10022  015206
ERR038  006740 G        HC.BF1= 000100          I$PWR = 000041          L$DU    021312 G        L10023  017762
ERR040  006752 G        HC.BF2= 000206          I$RPT = 000041          L$DUT   002072 G        L10025  021300
EVL   = 000004 G        HC.BSZ= 000106          I$SEG = 000041          L$DVTY  002374 G        L10026  021302
E$END = 002100          HC.CCT= 000012          I$SETU= 000041          L$EF    002052 G        L10027  021310
E$LOAD= 000035          HC.CEV= 000014          I$SFT = 000041          L$ENVI  002044 G        L10030  021316
FFREE   002156 G        HC.CMD= 000010          I$SRV = 000041          L$ERRT  002146 G        L10031  021324
FMEM    002162          HC.CPK= 000020          I$SUB = 000041          L$ETP   002102 G        L10032  025710
FMEMS   002164          HC.ESZ= 000004          I$TST = 000041          L$EXP1  002046 G        L10033  025744
FMERR   007320          HC.INT= 000000          J$JMP = 000167          L$EXP4  002064 G        L10034  026124
FRMTT   002541          HC.ISZ= 000004          KW.BRL  002322          L$EXP5  002066 G        L10035  000050R 003
FSIZE   002160 G        HC.MCT= 000006          KW.CSR  002320          L$HARD  025714 G        L10037  000056R 003
F$AU  = 000015          HC.MEV= 000014          KW.EL   002330          L$HIME  002120 G        MD.CMP= 040000
F$AUTO= 000020          HC.MPK= 000020          KW.HZ   002326          L$HPCP  002016 G        MD.CWB= 000010
F$BGN = 000040          HC.MSG= 000004          KW.OUT= 000105 G        L$HPTP  002022 G        MD.ERR= 010000
F$CLEA= 000007          HC.PSZ= 000060          KW.VEC  002324          L$HW    002130 G        MD.EXP= 100000
F$DU  = 000016          HC.RSZ= 000004          KW11I   015166 G        L$ICP   002104 G        MD.FEU= 000001
F$END = 000041          HC.SIZ= 000314          LBSIZ = 001060          L$INIT  017772 G        MD.IMF= 000002
F$HARD= 000004          HM.CYL= 020000 G        LBUFE   002372          L$LADP  002026 G        MD.NXU= 000001
F$HW  = 000013          HOE   = 100000 G        LBUFN   002370          L$LAST  000044RG  003  MD.PRI= 000001
F$INIT= 000006          H.DRV = 000002          LBUFS   002366          L$LOAD  002100 G        MD.RIP= 000001
F$JMP = 000050          H.PRM = 000004          LDDM    010322          L$LUN   002074 G        MD.SCH= 004000
F$MOD = 000000          H.UBA = 000000          LDNEXT  010354          L$MREV  002050 G        MD.SCL= 002000
F$MSG = 000011          IBE   = 010000 G        LF    = 000012 G        L$NAME  002000 G        MD.SEC= 000100
F$PROT= 000021          ICONT = 000002 G        LOADB   014460          L$PRIO  002042 G        MD.SEQ= 000020
F$PWR = 000017          IDU   = 000040 G        LOADDM  014376          L$PROT  017764 G        MD.SER= 000400
F$RPT = 000012          IER   = 020000 G        LOADER  014566          L$PRT   002112 G        MD.SPD= 000001
F$SEG = 000003          IFLAGS  002200 G        LOADE1  014556          L$REPP  002062 G        MD.SSH= 000200
F$SOFT= 000005          INITBL  002346          LOADTX  014454          L$REV   002010 G        MD.SWP= 000004
F$SRV = 000010          INITWA  003246          LOE   = 040000 G        L$RPT   016430 G        MD.VOL= 000002
F$SUB = 000002          INITWB  003356          LOG   = 000001          L$SOFT  026050 G        MD.WBN= 000100
F$SW  = 000014          INITWC  002465          LOGCHK  021426          L$SPC   002056 G        MD.WBV= 000400
F$TEST= 000001          INITXX  021232          LOGM1   003042          L$SPCP  002020 G        MEMFIL  011174
GETCDN  013360          INIT1   020264          LOGM2   003074          L$SPTP  002024 G        MESSG   003403
GETCNT  013314          INIT2   020446          LOGM3   003121          L$STA   002030 G        MLDRER  021244
GETCNX  013320          INIT3   020504          LOGOUT  021374          L$SW    002140 G        MSCP  = 000000
GETCXX  013366          INIT4   020540          LOT   = 000010 G        L$TEST  002114 G        MSGCST  026006
GTDRVT  013224          INIT5   021050          LPNT    007630          L$TIML  002014 G        MSGLDR  025776
G$CNTO= 000200          INIT6   021216          LPNTB   007602          L$UNIT  002012 G        MSGPKL  007104
G$DELM= 000372          INP28A  025557          LPNTF   007572          L10000  002136          MSGPKT  007060
G$DISP= 000003          INP28B  025621          LPNTS   007622          L10001  002146          MSGUBA  025744
G$EXCP= 000400          INTRCV  002212          LPNTX   007612          L10002  006554          MXFERE  012114
G$HILI= 000002          IPADRS  002244          L$ACP   002110 G        L10003  006572          MXFERP  002672
G$LOLI= 000001          IREST = 000004 G        L$APT   002036 G        L10004  006604          MXFERX  012112
G$NO  = 000000          ISR   = 000100 G        L$AU    021320 G        L10005  006616          NCON    007304
G$OFFS= 000400          ISTRT = 000010 G        L$AUT   002070 G        L10006  006634          NCONF   007730
G$OFSI= 000376          ISTRTH= 000020 G        L$AUTO  021302 G        L10007  006650          NCONS   007706
G$PRMA= 000001          IXE   = 004000 G        L$CCP   002106 G        L10010  006662          NOCLOC  003456
```

Symbol table

```
NXMAD   002340          PRI04 = 000200 G        RESPCT  010406          SA.MCV= 000017          SVCSUB= 000000
NXMI    015146 G        PRI05 = 000240 G        RESPDM  010374          SA.MP = 000100          SVCTAG= 000000
OP.ABO= 000001          PRI06 = 000300 G        RG.FLG= 040000          SA.MSE= 000007          SVCTST= 000000
OP.ACC= 000020          PRI07 = 000340 G        RG.OWN= 100000          SA.MSG= 003400          S$LSYM= 010000
OP.AVA= 000100          PS      007546          RNTIM   002347          SA.MS1= 000400          S.EL    026211
OP.AVL= 000010          PTYPE   002342          RNTIME  016236          SA.NV = 002000          S.IW    026345
OP.CCD= 000021          PX      007522          RNTIMX  016416          SA.NVE= 000400          S.LOG   026377
OP.CMP= 000040          P.BCNT= 000014          RNTIM1  002572          SA.PRG= 000001          S.MAN   026124
OP.DUP= 000101          P.BUFF= 000020          RNTIM2  002600          SA.QB = 001000          S.SSF   026307
OP.ELP= 000003          P.CHVR= 000023          RPTCT   016524          SA.SM = 000040          S.XL    026225
OP.END= 000200          P.CMST= 000020          RPTCTN  017152          SA.STE= 000200          TEMP    002214
OP.ERS= 000022          P.CNTF= 000016          RPTDT   016564          SA.STP= 100000          TEST4 = ****** GX
OP.ESP= 000002          P.CNTI= 000024          RPTDTN  017146          SA.S1 = 004000          TINDEX= 000006
OP.FLU= 000023          P.CPSP= 000042          RPTMD2  017527          SA.S2 = 010000          TINIT   010234
OP.GCS= 000002          P.CRF = 000000          RPTMD3  017551          SA.S3 = 020000          TNAMES  010200
OP.GSS= 000001          P.CSVR= 000022          RPTMSD  017460          SA.S4 = 040000          TNUM    002202
OP.GUS= 000003          P.CTMO= 000020          RPTMSG  017206          SA.TST= 000000          TOOMER  021262
OP.MRD= 000030          P.CYLS= 000050          RPTMSH  017242          SA.VCE= 000177          TSTTAB  002174
OP.MWR= 000031          P.DEXT= 000014          RPTMS4  017631          SA.VEC= 000177          TTYOUT  002344
OP.ONL= 000011          P.DFLG= 000017          RPTMS5  017702          SA.WRP= 040000          T$ARGC= 000001
OP.RD = 000041          P.DMDT= 000024          RPTXX   017166          SEKERE  011752          T$CODE= 002130
OP.RLC= 000103          P.DPRG= 000020          RSPDRP  010604          SET00   015222          T$ERRN= 000003
OP.RPL= 000024          P.DTMO= 000024          RSPDSP  011126          SET01   015230          T$EXCP= 000000
OP.RSD= 000005          P.ELGF= 000034          RSPERR  010670          SET02   015236          T$FLAG= 000040
OP.SCC= 000004          P.FBBK= 000034          RSPIN   010622          SETTO   015210          T$FREE= 000056R     003
OP.SEX= 000007          P.FLGS= 000011          RSPMWR  010642          SFPTBL  002140 G        T$GMAN= 000000
OP.SMC= 000102          P.GRPS= 000046          RSPNRP  010570          SM.IW = 040000 G        T$HILI= 177777
OP.SSD= 000004          P.HSTI= 000020          RSPNTO  010534          SM.LOG= 001000 G        T$LAST= 000001
OP.SUC= 000012          P.HTMO= 000020          RSPNXT  010536          SM.MAN= 000200 G        T$LOLI= 000000
OP.WR = 000042          P.LBN = 000034          RSPOU   010710          SM.SSF= 000400 G        T$LSYM= 010000
OSTRE   007744          P.MEDI= 000034          RSPOUT  011024          SNDCMD  014664          T$LTNO= 000001
OSTRNG  007676          P.MLUN= 000014          RSPOU2  011072          SND.S1  002350          T$NEST= 177777
O$APTS= 000000          P.MOD = 000012          RSPOU3  011100          SND.S2  002354          T$NSO = 000005
O$AU  = 000000          P.OPCD= 000010          RSPPTM  010702          SND.S3  002360          T$PCNT= 000000
O$BGNR= 000001          P.OTRF= 000014          RSPPT2  010712          SO.BIT= 000004          T$PTAB= 010036
O$BGNS= 000001          P.OVRL= 000034          RSPPT3  010762          SO.EL = 000000          T$PTHV= 000001
O$DU  = 000000          P.RBN = 000014          RSPRPT  010566          SO.XL = 000002          T$PTNU= 000001
O$ERRT= 000001          P.RBNS= 000056          RSPTM   010472          SSTEP4  002362          T$SAVL= 177777
O$GNSW= 000001          P.RCTC= 000057          RSPTMO  010522          STIME   002334          T$SEGL= 177777
O$POIN= 000001          P.RCTS= 000054          RSP.CK  015756          ST.ABO= 000002          T$SIZE= 000005
O$SETU= 000001          P.RGID= 000034          RSP.S1  015710          ST.ADL= 000400          T$SUBN= 000000
PAT16C  002256          P.RGOF= 000040          RSP.S2  015722          ST.AVL= 000004          T$TAGL= 177777
PAT16W  002260          P.SHST= 000042          RSP.S3  015742          ST.CMD= 000001          T$TAGN= 010040
PB      007476          P.SHUN= 000040          RUNDM   010300          ST.CMP= 000007          T$TEMP= 000005
PF      007452          P.STS = 000012          SA.BST= 000374          ST.CNT= 000012          T$TEST= 000001
PNT   = 001000 G        P.TIME= 000024          SA.CMD= 034000          ST.DAT= 000010          T$TSTM= 177777
PNTERR  014232          P.TRKS= 000044          SA.CME= 000070          ST.DIA= 000037          T$TSTS= 000001
PNTNUM  013372          P.UADR= 000020          SA.CM1= 004000          ST.DRV= 000013          T$$AU = 010031
PNTNUS  013400          P.UNFL= 000016          SA.CNT= 000360          ST.HST= 000011          T$$AUT= 010026
PNTPKL  007016          P.UNIT= 000004          SA.CTP= 003400          ST.MFE= 000005          T$$CLE= 010027
PNTPKT  006766          P.UNSZ= 000044          SA.DI = 000400          ST.MSK= 000037          T$$DAT= 010037
PRI   = 002000 G        P.UNTI= 000024          SA.ERC= 003777          ST.OFL= 000003          T$$DU = 010030
PRINTC  007416          P.USEF= 000022          SA.ERR= 100000          ST.SUB= 000040          T$$HAR= 010033
PRI00 = 000000 G        P.VRSN= 000014          SA.GO = 000001          ST.SUC= 000000          T$$HW = 010000
PRI01 = 000040 G        P.VSER= 000050          SA.INE= 000200          ST.WPR= 000006          T$$INI= 010025
PRI02 = 000100 G        READDT  010276          SA.INT= 000200          SVCGBL= 000000          T$$MSG= 010017
PRI03 = 000140 G        RESET   016134          SA.LFC= 000002          SVCINS= 000000          T$$PC = 000001
```

Symbol table

```
T$$PRO= 010024          T4DPD   025115          T4Q07   022444          T4SEK   024555          XFCE    005725
T$$PTA= 010036          T4ECC   024422          T4Q08   022522          T4STRT  021464          XFRU    006471
T$$RPT= 010023          T4END   024645          T4Q09   022530          T4TRAK  024710          XMSG1   005777
T$$SOF= 010034          T4EXIT  021772          T4Q10   022620          T4TRC   024657          XMSG2   006033
T$$SRV= 010022          T4GRC   024716          T4Q11   022664          T4WAIT  021756          XPKT1   006100
T$$SW = 010001          T4GRP   024747          T4Q12   023062          T4WARN  003150          XPKT2   006405
T$$TES= 010032          T4OPT1  025247          T4Q13   023116          T4WCA   024257          XSA     006434
T1      021326 G        T4OPT2  025306          T4Q14   023150          T4WCR   024313          X$ALWA= 000000
T4BB    024122          T4OPT3  025342          T4Q15   023176          T4WO    024244          X$FALS= 000040
T4BBI   024220          T4OPT4  025405          T4Q16   023212          UAM   = 000200 G        X$OFFS= 000400
T4BE    024600          T4OPT5  025454          T4Q17   023220          UCNT    002210          X$TRUE= 000020
T4BEG   024631          T4OPT6  025523          T4Q18   023232          UF.CMR= 000001          X1A     003666
T4CON   021446          T4OPT7  002462          T4Q19   023240          UF.CMW= 000002          X14     004336
T4CYL   024755          T4PRM1  021530          T4Q20   023316          UF.INA= 040000          X2      003751
T4CYLB  025010          T4PRM2  021542          T4Q21   023350          UF.RPL= 100000          X2A     003666
T4CYLE  025032          T4PRM3  021576          T4Q22   023410          UF.SCH= 004000          X3      004020
T4DCA   024455          T4PRM4  021602          T4Q23   023476          UF.SCL= 002000          X3A     003666
T4DCR   024503          T4PRM5  024064          T4Q24   023534          UF.WBN= 000100          X31     004737
T4DEF   021634          T4QHED  025127          T4Q25   023566          UF.WPH= 020000          X32     005061
T4DEFA  021640          T4QUES  022000          T4Q26   023622          UF.WPS= 001000          X35     005172
T4DEFB  021652          T4QUE2  024014          T4Q27   023702          UF.576= 000004          X36     005252
T4DEFC  021676          T4QU2E  024120          T4Q27A  023714          URNING  002206          X38     005377
T4DEFD  021704          T4Q01   022104          T4Q28   023722          URUN    002204          X4      004106
T4DEFE  021710          T4Q02   022142          T4Q29   023754          UTOT1   012130          X40     005603
T4DEFW  021624          T4Q03   022240          T4Q30   024006          UTOT1A  012162          X6      004251
T4DMN   024147          T4Q04   022316          T4RET   024536          UTOT2   012166          X8A     003666
T4DP    024354          T4Q05   022324          T4RO    024232          WAITMS  014774          $PATCH  000000RG    003
T4DPC   025052          T4Q06   022414          T4RUN   021720

. ABS.  026420    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
DM      000000    002   (RW,I,LCL,ABS,CON)
END     000056    003   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 732
Work  file writes: 625
Size of work file: 28769 Words  ( 113 Pages)
Size of core pool: 14336 Words  ( 56 Pages)
Operating  system: RT-11 (Under RTEM-11)

Elapsed time: 00:04:51.00
ZUDIAO,ZUDIAO/C=SVC34R.MLB,ZUDIAO.DOC,CZUDHO.MAC
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
BIT6     91-8#     97-19     98-23
BIT7     91-8#     98-31    120-56
BIT8     91-8#     98-22
BIT9     91-8#     98-21
BLD28   126-8#    151-25    151-142   151-146   151-195   151-199
BLDC0   130-26    130-28#
BLDC1   130-30#   130-32
BLDCMD  105-116   105-134   129-66    130-17#
BOE      91-8#
BRSAV    99-41#   143-208*  143-248   143-250*
C$AU     87-153#  148-14
C$AUTO   87-153#  145-12
C$BRK    87-153#  105-47    133-27    137-17    138-30    150-114
C$BSEG   87-153#
C$BSUB   87-153#
C$CEFG   87-153#
C$CLCK   87-153#  143-127   143-129
C$CLEA   87-153#  146-16
C$CLOS   87-153#
C$CLP1   87-153#
C$CVEC   87-153#  137-33
C$DCLN   87-153#  102-11    143-105   143-337   144-21    144-30
C$DODU   87-153#
C$DRPT   87-153#  105-59    117-56    128-26    150-117
C$DU     87-153#  147-12
C$EDIT   87-153#   87-202
C$ERDF   87-153#  105-26    105-44    105-94    105-112   121-25    129-91    133-35    133-48    136-122   137-34    137-95    138-42    138-53
C$ERHR   87-153#
C$ERRO   87-153#  128-41
C$ERSF   87-153#  102-9     136-128   136-134   143-103   144-19    144-28
C$ERSO   87-153#
C$ESCA   87-153#
C$ESEG   87-153#
C$ESUB   87-153#
C$ETST   87-153#  152-47
C$EXIT   87-153#  143-353   146-14    150-118
C$GETB   87-153#
C$GETW   87-153#
C$GMAN   87-153#  143-333   151-18    151-26    151-33    151-39    151-45    151-46    151-49    151-54    151-67    151-68    151-71    151-74
        151-76    151-100   151-137   151-143   151-147   151-162   151-166   151-172   151-176   151-182   151-196   151-200   151-213   151-219
C$GPHR   87-153#  143-93    143-209
C$GPLO   87-153#
C$GPRI   87-153#
C$INIT   87-153#  144-42
C$INLP   87-153#
C$MANI   87-153#  143-331   150-42
C$MEM    87-153#  143-146
C$MSG    87-153#  101-17    101-21    101-25    101-29    101-39    101-102   101-106   101-111   101-115   101-120   101-124   101-128   101-138
        101-186
C$OPEN   87-153#
C$PNTB   87-153#  102-80
C$PNTF   87-153#  102-78    127-39    127-43    143-320   151-16    151-94    151-95    151-96    151-97    151-98    151-99
C$PNTS   87-153#  102-84    141-26    141-36    141-59    141-61    141-82
C$PNTX   87-153#  102-82
C$QIO    87-153#
C$RDBU   87-153#
```

Cross reference table (CREF V05.01)

```
C$REFG   87-153# 143-46   143-52   143-58   143-65
C$RESE   87-153#  87-153#
C$REVI   87-153#  87-202
C$RFLA   87-153# 114-35   116-40   117-45
C$RPT    87-153# 141-107
C$SEFG   87-153#
C$SPRI   87-153# 143-351
C$SVEC   87-153# 129-20   137-30   139-14   143-140
C$TPRI   87-153#
C.DR0    97-37#  115-19   121-16   141-29   143-77   143-267  143-314  150-51   150-84
C.DR1    97-38#
C.DR2    97-39#
C.DR3    97-40#
C.FLG    97-35#  104-24*  105-13   105-16   105-18   105-70*  105-79   105-99   105-130* 105-131  105-137* 105-140* 114-39*  116-51*
        118-19*  129-83*  129-84*  131-19*  133-18   133-38*  143-256
C.HCOM   97-36#  101-156  102-54*  105-12   130-18   131-15
C.JAD    97-34#
C.JSR    97-33#  129-19
C.REF    97-43#  105-92   131-14*  131-15
C.SIZE   97-45#  104-32   105-63   141-86   143-84   143-99   143-220  143-236  143-256  143-323  150-63   150-96
C.TO     97-41#  105-42   105-145  133-15   133-33   138-25   138-36
C.TOH    97-42#  105-39   133-30   138-33
C.UADR   97-30#  101-173  137-28   143-215
C.UNIT   97-31#  104-25   104-26   105-15   128-17   136-121* 141-24   143-75*  143-110*
C.VEC    97-32#  129-16
CALR1   102-254  120-9#   120-57
CALR2   102-255  120-24#
CALR3   102-256  120-56#
CALR4   102-257  120-68#
CALR5   102-258  120-81#
CALR6   102-259  120-93#
CALR7   102-260  120-105#
CALR8   102-261  120-117#
CALR9   102-262  120-130#
CF.576   94-94#
CF.ATN   94-89#
CF.MSC   94-90#
CF.OTH   94-91#
CF.SHD   94-93#
CF.THS   94-92#
CLRBFL  132-23#  132-25
CLRBUF  105-118  105-136  132-17#
CNTINT  129-23   136-30#
CNTIST  136-67   137-16#
CNTRSD   99-98#  137-44*  137-55*  137-56   138-22*  138-28
CNTRSP  136-81   137-51   138-21#
CNTSRV  134-32#  143-254
CNV28   127-15#  151-27   151-144  151-148  151-197  151-201
CON.A   102-181# 102-297
CON.A1  102-182# 102-184  120-39
CON.A2  102-186  102-188#
CON.D   102-194# 102-298
CON.H   102-202# 102-299
CON.N   102-218# 102-301
CON.N1  102-219# 102-221
CON.O   102-210# 102-300
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
F$HARD   87-1530 153-11   153-41
F$HW     87-1530  89-12    89-23
F$INIT   87-1530 143-44   144-42
F$JMP    87-1530 141-93   141-93   143-353  146-14   147-10   147-10   148-12   148-12   150-118
F$MOD    87-1530
F$MSG    87-1530 101-15   101-17   101-19   101-21   101-23   101-25   101-27   101-29   101-37   101-39   101-100  101-102  101-104
       101-106  101-108  101-111  101-113  101-115  101-117  101-120  101-122  101-124  101-126  101-128  101-136  101-138  101-165
       101-186
F$PROT   87-1530 142-9    142-15
F$PWR    87-1530
F$RPT    87-1530 141-10   141-107
F$SEG    87-1530
F$SOFT   87-1530 154-12   154-55
F$SRV    87-1530 134-11   134-13   134-32   134-35   134-50   134-54
F$SUB    87-1530
F$SW     87-1530  90-11    90-21
F$TEST   87-1530 150-3    152-47
FFREE    99-100  102-30   102-34*  103-21*  107-30   107-42   107-46   137-21   143-146* 143-147  143-154  143-168  143-344
FMEM     99-120  103-21   143-344*
FMEMS    99-130  103-22   143-345*
FMERR   102-90   102-32   136-52
FRMTT   100-350  102-70
FSIZE    99-110  102-31*  103-22*  107-31   107-44   136-50   143-147* 143-345
G$CNTO   87-1530
G$DELM   87-1530
G$DISP   87-1530
G$EXCP   87-1530
G$HILI   87-1530
G$LOLI   87-1530
G$ND     87-1530 143-333
G$OFFS   87-1530 143-333  151-18   151-26   151-33   151-39   151-45   151-46   151-49   151-54   151-67   151-68   151-71   151-74
       151-76   151-100  151-137  151-143  151-147  151-162  151-166  151-172  151-176  151-182  151-196  151-200  151-213  151-219
       153-27   153-35   153-38   154-37   154-39   154-42   154-44   154-46   154-48
G$OFSI   87-1530 143-333  151-18   151-26   151-33   151-39   151-45   151-46   151-49   151-54   151-67   151-68   151-71   151-74
       151-76   151-100  151-137  151-143  151-147  151-162  151-166  151-172  151-176  151-182  151-196  151-200  151-213  151-219
       153-27   153-35   153-38   154-37   154-39   154-42   154-44   154-46   154-48
G$PRMA   87-1530 153-27
G$PRMD   87-1530 151-18   151-26   151-54   151-100  151-137  151-143  151-147  151-162  151-166  151-172  151-176  151-196  151-200
       151-213  151-219  153-35   154-39   154-42
G$PRML   87-1530 143-333  151-33   151-39   151-45   151-46   151-49   151-67   151-68   151-71   151-74   151-76   151-182  153-38
       154-37   154-44   154-46   154-48
G$RADA   87-1530 151-26   151-143  151-147  151-196  151-200
G$RADB   87-1530
G$RADD   87-1530 151-18   151-54   151-100  151-137  151-162  151-166  151-172  151-176  151-213  153-35   154-39   154-42
G$RADL   87-1530 143-333  151-33   151-39   151-45   151-46   151-49   151-67   151-68   151-71   151-74   151-76   151-182  153-38
       154-37   154-44   154-46   154-48
G$RADO   87-1530 151-219  153-27
G$XFER   87-1530
G$YES    87-1530 151-18   151-26   151-33   151-39   151-45   151-46   151-49   151-54   151-67   151-68   151-71   151-74   151-76
       151-100  151-137  151-143  151-147  151-162  151-166  151-172  151-176  151-182  151-196  151-200  151-213  151-219  153-27
       153-35   153-38   154-37   154-39   154-42   154-44   154-46   154-48
GETCDN  122-17   122-19   122-290
GETCNT  102-181  102-218  102-229  102-244  122-140  123-16
GETCNX  122-160  122-28
GETCXX  122-30   122-320
GTDRVT  109-60   110-24   111-17   112-28   113-15   114-21   117-37   118-22   121-140  128-21   143-108
```

Cross reference table (CREF V05.01)

```
H.DRV   143-107  143-271  143-281  153-19#  153-35   153-35   153-35
H.PRM   143-284  153-21#  153-38   153-38   153-38
H.UBA   143-216  153-14#  153-27   153-27   153-27
HC.BF1   93-68#   93-69   105-117
HC.BF2   93-69#   93-71   101-157  105-104  105-108*  105-109  105-135
HC.BSZ   93-52#   93-69    93-71   105-120  116-72    116-76   132-20   132-22   150-20
HC.CCT   93-60#  131-17*
HC.CEV   93-65#  130-20
HC.CMD   93-59#   93-60    93-62   102-59*
HC.CPK   93-66#   93-68   101-143  102-58   129-67*   129-68*  129-69*  129-70*  130-33*  131-15*  132-19*  132-20*
HC.ESZ   93-50#   93-63
HC.INT   93-54#   93-56
HC.ISZ   93-48#   93-56
HC.MCT   93-57#  131-16*
HC.MEV   93-62#   93-63    93-65
HC.MPK   93-63#   93-66   101-145  102-56   102-58    105-82   105-87   105-92   129-81
HC.MSG   93-56#   93-57    93-59   102-57*  137-22
HC.PSZ   93-51#   93-68   130-21   130-29
HC.RSZ   93-49#   93-59    93-62
HC.SIZ   93-71#  102-51
HCOMM   102-51#  129-25
HM.CYL  143-287  143-309  153-24#  153-38
HOE      91-8#
I$AU     87-153#  148-9#   148-14#
I$AUTO   87-153#  145-10#  145-12#
I$CLN    87-153#  146-8#   146-14   146-16#
I$DU     87-153#  147-8#   147-12#
I$HRD   153-11#  153-41#
I$INIT   87-153#  143-44#  143-353  144-42#
I$MOD    87-153#
I$MSG    87-153#  101-15#  101-17#  101-19#  101-21#  101-23#  101-25#  101-27#  101-29#  101-37#  101-39#  101-100#  101-102#  101-104#
         101-106#  101-108#  101-111#  101-113#  101-115#  101-117#  101-120#  101-122#  101-124#  101-126#  101-128#  101-136#  101-138#  101-165#
         101-186#
I$PROT   87-153#  142-9#
I$PTAB   87-153#  156-14   156-14#  156-24   156-24#
I$PWR    87-153#
I$RPT    87-153#  141-10#  141-107#
I$SEG    87-153#  150-3
I$SETU   87-153#  156-13   156-13#  156-14   156-25   156-25#
I$SFT   154-12#  154-55#
I$SRV    87-153#  134-11#  134-13#  134-32#  134-35#  134-50#  134-54#
I$SUB    87-153#  150-3
I$TST    87-153#  150-3    150-3#   150-118  152-47   152-47#  152-47#
IBE      91-8#
ICONT    99-22#  143-62   150-11
IDU      91-8#   114-36   116-41   117-46
IER      91-8#
IFLAGS   99-20#  109-70   143-49*  143-55*  143-61*  143-62*  150-11   150-13*  150-38   150-102*  150-103*
INIT1   143-50   143-56   143-125#
INIT2   143-154#
INIT3   143-168#
INIT4   143-181#
INIT5   143-309#
INIT6   143-310  143-332  143-335  143-344#
INITBL   99-89#  137-45
INITWA  100-48#  143-311
```

Cross reference table (CREF V05.01)

```
INITWB  100-49#  143-320
INITWC  100-28#  143-333
INITXX  143-118  143-351#
INP28A  127-39   152-40#
INP28B  127-43   152-41#
INTRCV   99-31#
IPADRS   99-35#  139-16   143-171  143-227  143-228
IREST    99-23#  143-55   150-38
ISR      91-8#
ISTRT    99-24#  143-49   150-38
ISTRTH   99-25#  109-70   143-61   150-103
IXE      91-8#
J$JMP    87-153# 141-93   147-10   148-12
KW.BRL   99-66#  143-136*
KW.CSR   99-65#  105-37   105-51   133-28   134-53*  138-31   139-22   139-24*  140-14   143-131* 143-135* 143-141*
KW.EL    99-69#  105-39   105-42   105-53   105-56   133-30   133-33   134-51*  134-52*  135-29   135-30   135-31   138-33   138-36
        140-17   140-18   143-125* 143-126*
KW.HZ    99-68#  135-16   140-19   143-138*
KW.OUT   91-13   134-53   139-24   143-141
KW.VEC   99-67#  143-137* 143-140
KW11I   134-50#  143-140
L$ACP    87-202#
L$APT    87-202#
L$AU    148-9#
L$AUT    87-202#
L$AUTO   87-202  145-10#
L$CCP    87-202#
L$CLEA   87-202  146-8#
L$CO     87-202#
L$DEPO   87-202#
L$DESC   87-202  100-20#
L$DESP   87-202#
L$DEVP   87-202#
L$DISP   87-202   88-11#
L$DLY    87-202#
L$DTP    87-202#
L$DTYP   87-202#
L$DU    147-8#
L$DUT    87-202#
L$DVTY   87-202  100-13#
L$EF     87-202#
L$ENVI   87-202#
L$ERRT   87-202   99-8#
L$ETP    87-202#
L$EXP1   87-202#
L$EXP4   87-202#
L$EXP5   87-202#
L$HARD   87-202  153-11   153-11#
L$HIME   87-202# 107-55
L$HPCP   87-202#
L$HPTP   87-202#
L$HW     87-202   89-12    89-12#
L$ICP    87-202#
L$INIT   87-202  143-44#
L$LADP   87-202#
L$LAST   87-202  155-17#  156-25
```

Cross reference table (CREF V05.01)

```
L$LOAD   87-202#
L$LUN    87-202# 104-25*  105-15*  121-31*  128-17*
L$MREV   87-202#
L$NAME   87-202#
L$PRIO   87-202#
L$PRJT   87-202  142-9#
L$PRT    87-202#
L$REPP   87-202#
L$REV    87-202#
L$RPT    87-202  141-10#
L$SOFT   87-202  154-12   154-12#
L$SPC    87-202#
L$SPCP   87-202#
L$SPTP   87-202#
L$STA    87-202#
L$SW     87-202   90-11    90-11#
L$TEST   87-202#
L$TIML   87-202#
L$UNIT   87-202# 143-113  143-156  143-299
L10000   89-12    89-23#
L10001   90-11    90-21#
L10002  101-17#
L10003  101-21#
L10004  101-25#
L10005  101-29#
L10006  101-39#
L10007  101-102#
L10010  101-106#
L10011  101-111#
L10012  101-115#
L10013  101-120#
L10014  101-124#
L10015  101-128#
L10016  101-138#
L10017  101-186#
L10020  134-13#
L10021  134-35#
L10022  134-54#
L10023  141-93   141-107#
L10025  143-353  144-42#
L10026  145-12#
L10027  146-14   146-16#
L10030  147-10   147-12#
L10031  148-12   148-14#
L10032  150-118  152-47#
L10033  153-11   153-41#
L10034  154-12   154-55#
L10035  156-14#
L10037  156-14   156-24#
LBSIZ    99-116# 116-65   116-69
LBUFE    99-115# 116-70*  116-73
LBUFN    99-114# 116-68*  116-71   116-72*  116-73   116-81*  150-21
LBUFS    99-113# 103-19*  116-63   116-67*  150-14   150-17*
LDDM    104-23#  104-34
LDNEXT  104-27   104-30   104-32#
LF       91-10
```

Cross reference table (CREF V05.01)

```
LOADB   129-65#
LOADDM  104-29   129-15#
LOADE1  129-82   129-91#
LOADER  129-24   129-80   129-92#
LOADTX  129-64#
LOE      91-8#
LOG      93-15#
LOGCHK  150-15   150-26#
LOGM1   100-44#  150-16
LOGM2   100-45#  150-23
LOGM3   100-46#  150-28
LOGOUT  150-18#  150-22
LOT      91-8#
LPNT    102-94   102-97   102-100  102-104# 140-25   140-29   140-34
LPNTB   101-16   101-20   101-24   101-28   101-38   101-101  101-105  101-109  101-118  101-123  101-127  101-137  101-141  101-148
        101-155  101-159  101-172  102-96#  120-68   120-81
LPNTF   102-93#  102-156  102-236  143-132  143-311  150-16   150-23   150-28   150-77
LPNTS   102-102# 141-13   141-21
LPNTX   102-99#  114-40   114-42   117-53   118-26
MD.CMP   94-55#
MD.CWB   94-73#
MD.ERR   94-57#
MD.EXP   94-56#
MD.FEU   94-67#
MD.IMF   94-71#
MD.NXU   94-69#
MD.PRI   94-74#
MD.RIP   94-70#
MD.SCH   94-58#
MD.SCL   94-59#
MD.SEC   94-60#
MD.SEQ   94-65#
MD.SER   94-61#
MD.SPD   94-66#
MD.SSH   94-62#
MD.SWP   94-72#
MD.VOL   94-68#
MD.WBN   94-63#
MD.WBV   94-64#
MEMFIL  107-35#  107-38
MESSG   100-66#  114-40   118-26
MLDRER  143-273  144-18#
MSCP     93-14#
MSGCST  153-38   153-50#
MSGLDR  153-35   153-48#
MSGPKL  101-159# 101-162
MSGPKT  101-110  101-119  101-155#
MSGUBA  153-27   153-43#
MXFERE  114-22   114-44#
MXFERP  100-41#  114-42
MXFERX  114-27   114-32   114-34   114-37   114-43#
NCON    101-182  101-184#
NCONF   102-153  102-158#
NCONS   102-152# 102-155
NOCLOC  100-67#  143-132
NODOCU    2-39
```

```
NXMAD    99-72#  134-12*  137-29*  137-34   139-12*
NXMI    134-11#  137-30   139-14
O$APTS   87-153#  87-202
O$AU     87-153#  87-202
O$BGNR   87-153#  87-192#  87-202
O$BGNS   87-153#  87-192#  87-202
O$DU     87-153#  87-202
O$ERRT   87-153#  87-192#  87-202
O$GNSW   87-153#  87-192#  87-202
O$POIN   87-153#  87-192   87-192#  87-192#  87-192#  87-192#  87-192#  87-202
O$SETU   87-153#  87-192#  87-202  155-17
OP.ABO   94-20#
OP.ACC   94-21#
OP.AVA   94-39#
OP.AVL   94-22#
OP.CCD   94-23#
OP.CMP   94-24#
OP.DUP   94-40#
OP.ELP   94-46#
OP.END   94-37#  105-78   105-81
OP.ERS   94-25#
OP.ESP   94-45#  129-65
OP.FLU   94-26#
OP.GCS   94-27#
OP.GSS   94-44#
OP.GUS   94-28#
OP.MRD   94-35#  130-25
OP.MWR   94-36#  130-23
OP.ONL   94-29#
OP.RD    94-30#
OP.RLC   94-42#
OP.RPL   94-31#
OP.RSD   94-48#  105-81   105-133
OP.SCC   94-32#
OP.SEX   94-38#
OP.SHC   94-41#
OP.SSD   94-47#  105-78   105-115
OP.SUC   94-33#
OP.WR    94-34#
OSTRE   102-150  102-157  102-162#
OSTRNG  101-185  102-109  102-149#  102-161  118-36   120-14   120-95   120-107  120-119
P.BCNT   95-11#   96-12#  129-68*   132-20*
P.BUFF   95-12#
P.CHVR   96-63#
P.CMST   96-20#
P.CNTF   95-43#   96-60#
P.CNTI   96-64#
P.CPSP   95-30#
P.CRF    95-7#    96-7#   105-92    131-15*
P.CSVR   96-62#
P.CTMO   96-61#
P.CYLS   96-35#
P.DEXT   96-70#
P.DFLG   96-71#
P.DMDT   95-59#
P.DPRG   96-72#
```

Cross reference table (CREF V05.01)

```
P.DTMO  96-73#
P.ELGF  95-28#
P.FDBK  96-13#
P.FLGS  96-10#
P.GRPS  96-34#
P.HSTI  95-27#   96-28#   96-47#
P.HTMO  95-44#
P.LBN   95-14#
P.MEDI  96-30#   96-49#
P.MLUN  96-26#   96-45#
P.MOD   95-10#
P.OPCD  95-9#    96-9#    105-82   130-33*
P.OTRF  95-20#   96-19#
P.OVRL  95-60#  129-69*  129-70*
P.RBN   95-36#
P.RBNS  96-37#
P.RCTC  96-38#
P.RCTS  96-36#
P.RGID  95-52#
P.RGOF  95-53#
P.SHST  96-32#   96-51#
P.SHUN  95-29#   96-31#   96-50#
P.STS   96-11#  105-87   129-81
P.TIME  95-46#
P.TRKS  96-33#
P.UADR  95-13#  129-67*  132-19*
P.UNFL  95-26#   96-27#   96-46#
P.UNIT  95-8#    96-8#
P.UNSZ  96-52#
P.UNTI  96-29#   96-48#
P.USEF  95-45#
P.VRSN  95-42#   96-59#
P.VSER  96-53#
PAT16C  99-42#  108-20   150-69   150-72
PAT16W  99-43#  150-72
PB     102-80#  102-96
PF      99-80   102-78#  102-93
PNT     91-8#
PNTERR 116-56   128-14#  150-19
PNTNUM 102-195  102-203  102-211  123-15#
PNTNUS 120-31   123-17#
PNTPKL 101-148# 101-152
PNTPKT 101-114  101-141#
PRI     91-8#
PRI00   91-8#   143-351
PRI01   91-8#
PRI02   91-8#
PRI03   91-8#
PRI04   91-8#
PRI05   91-8#
PRI06   91-8#
PRI07   87-202   91-8#   129-20   137-30   139-14   143-140
PRINTC 101-175  102-68#  102-172  102-182  102-219  102-245  120-32  120-40  120-70  120-83  123-59  123-67  140-28  140-33
       140-36   141-15
PS     102-84#  102-102
PTYPE   99-80#  101-184* 102-75   102-93*  102-96*  102-99*  102-102* 120-71* 120-84*
```

```
PX      101-184  102-82#  102-99   120-71   120-84
READDT  103-24   103-27#
RESET   103-17   139-12#  143-145
RESPCT  105-12#  105-65
RESPDM  105-10#  105-66   150-31   150-110
RG.FLG   93-8#   131-16
RG.OWN   93-7#   131-16   131-17
RNTIM   100-37#  140-25
RNTIM1  100-38#  140-29
RNTIM2  100-39#  140-34
RNTIME  101-174  114-41   118-27   120-69   120-82   140-14#  141-14
RNTIMX  140-15   140-36#
RPTCT   141-24#  141-89
RPTCTN  141-32   141-86#
RPTDT   141-31#  141-85
RPTDTN  141-84#
RPTMD2  141-61   141-100#
RPTMD3  141-82   141-101#
RPTMS4  141-36   141-102#
RPTMS5  141-26   141-103#
RPTMSD  141-59   141-99#
RPTMSG  141-13   141-95#
RPTMSH  141-21   141-97#
RPTXX   141-88   141-91#
RSP.CK  137-70   137-79   137-90#
RSP.S1   99-89   137-68#
RSP.S2   99-91   137-75#
RSP.S3   99-93   137-84#
RSPDRP  105-27   105-45   105-70#  105-95   105-113  105-125
RSPDSP  105-124  105-151#  105-183
RSPERR  105-83   105-88   105-94#
RSPIN   105-17   105-78#
RSPMMR  105-80   105-82#
RSPNRP  105-52   105-55   105-57   105-63#
RSPNTO  105-38   105-41   105-43   105-46#
RSPNXT  105-14   105-51#  105-72   105-147
RSPOU   105-19   105-100#
RSPOU2  105-132  105-140#
RSPOU3  105-138  105-141#
RSPOUT  105-100  105-130#
RSPPT2  105-104#
RSPPT3  105-111  105-115#
RSPPTM  105-93   105-99#
RSPRPT  105-54   105-58#
RSPTM   105-25   105-31#
RSPTMO  105-40   105-44#
RUNDM   104-16#  150-108
S$LSYM   87-153#   89-23#   90-21#   101-17#  101-21#  101-25#  101-29#  101-39#  101-102#  101-106#  101-111#  101-115#  101-120#  101-124#
        101-128#  101-138#  101-186#  134-13#  134-35#  134-54#  141-107#  143-333  143-333  143-333  143-333#  144-42#  145-12#  146-16#
        147-12#  148-14#  151-18  151-18  151-18  151-18#  151-26  151-26  151-26  151-26#  151-33  151-33  151-33  151-33#
        151-39  151-39  151-39  151-39#  151-45  151-45  151-45  151-45#  151-46  151-46  151-46  151-46#  151-49  151-49
        151-49  151-49#  151-54  151-54  151-54  151-54#  151-67  151-67  151-67  151-67#  151-68  151-68  151-68  151-68#
        151-71  151-71  151-71  151-71#  151-74  151-74  151-74  151-74#  151-76  151-76  151-76  151-76#  151-100  151-100
        151-100  151-100#  151-137  151-137  151-137  151-137#  151-143  151-143  151-143  151-143#  151-147  151-147  151-147  151-147#
        151-162  151-162  151-162  151-162#  151-166  151-166  151-166  151-166#  151-172  151-172  151-172  151-172#  151-176  151-176
        151-176  151-176#  151-182  151-182  151-182  151-182#  151-196  151-196  151-196  151-196#  151-200  151-200  151-200  151-200#
```

Cross reference table (CREF V05.01)

	151-213	151-213	151-213	151-213#	151-219	151-219	151-219	151-219#	152-47#	153-41#	154-55#	
S.EL	154-39	154-62#										
S.IW	154-46	154-65#										
S.LOG	154-48	154-66#										
S.MAN	154-37	154-61#										
S.SSF	154-44	154-64#										
S.XL	154-42	154-63#										
SA.BST	92-86#											
SA.CM1	92-31#											
SA.CMD	92-26#											
SA.CME	92-56#											
SA.CNT	92-93#	136-116										
SA.CTP	92-59#											
SA.DI	92-39#	137-68										
SA.ERC	92-17#											
SA.ERR	92-7#	138-22										
SA.GO	92-84#	136-103										
SA.INE	92-76#											
SA.INT	92-24#											
SA.LFC	92-85#											
SA.MCV	92-92#	136-111										
SA.MP	92-40#	137-69										
SA.MS1	92-30#											
SA.MSE	92-55#											
SA.MSG	92-25#											
SA.NV	92-37#											
SA.NVE	92-77#											
SA.PRG	92-48#											
SA.QB	92-38#	137-69										
SA.S1	92-11#	137-44	137-68									
SA.S2	92-10#	137-78										
SA.S3	92-9#	137-86										
SA.S4	92-8#	137-56										
SA.SM	92-41#	137-69										
SA.STE	92-58#											
SA.STP	92-28#	137-19										
SA.TST	92-69#	137-23										
SA.VCE	92-75#											
SA.VEC	92-23#											
SA.WRP	92-27#											
SEKERE	113-16	113-19#										
SETOO	135-18#	135-23										
SETO1	135-19	135-21#										
SETO2	135-29#	135-32										
SETTO	105-146	133-16	135-14#	138-26	141-18	143-117	143-144					
SFPTBL	90-11#	109-72	114-31	114-33	116-48	116-53	117-43	150-26	150-40	150-111		
SM.IW	109-72	154-26#	154-46									
SM.LOG	116-53	150-26	150-111	154-25#	154-48							
SM.MAN	150-40	154-23#	154-37									
SM.SSF	116-48	154-24#	154-44									
SND.S1	99-90#	137-20*	137-75	137-84								
SND.S2	99-92#	137-21*	137-22*									
SND.S3	99-94#	137-23*										
SNDCMD	105-142	129-78	131-13#									
SO.BIT	109-72	116-48	116-53	150-26	150-40	150-111	154-18#	154-37	154-37	154-37	154-44	154-44	154-44	154-46
	154-46	154-46	154-48	154-48	154-48							

```
SO.EL   117-43   154-15#  154-39   154-39   154-39
SO.XL   114-31   114-33   154-16#  154-42   154-42   154-42
SSTEP4   99-97#  136-83*  136-109
ST.ABO   96-83#
ST.AOL   96-94#
ST.AVL   96-85#
ST.CMD   96-82#
ST.CMP   96-88#
ST.CNT   96-91#
ST.DAT   96-89#
ST.DIA   96-93#
ST.DRV   96-92#
ST.HST   96-90#
ST.MFE   96-86#
ST.MSK   96-79#  105-87   129-81
ST.OFL   96-84#
ST.SUB   96-80#
ST.SUC   96-81#
ST.WPR   96-87#
STIME    99-70#  105-53   105-56   141-16   143-115  143-142
SVCGBL   87-153#  87-162#  87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   88-11    88-11    88-11    89-12    89-12    89-12    89-12    89-12    89-12    90-11
         90-11    90-11    90-11    90-11    90-11    99-8     99-8     99-8     100-13   100-13   100-13   100-20   100-20   100-20
        101-15   101-15   101-15   101-19   101-19   101-19   101-23   101-23   101-23   101-27   101-27   101-27   101-37   101-37
        101-37   101-100  101-100  101-100  101-104  101-104  101-104  101-108  101-108  101-108  101-113  101-113  101-113  101-117
        101-117  101-117  101-122  101-122  101-122  101-126  101-126  101-126  101-136  101-136  101-136  101-165  101-165  101-165
        134-11   134-11   134-11   134-32   134-32   134-32   134-50   134-50   134-50   141-10   141-10   141-10   142-9    142-9
        142-9    143-44   143-44   143-44   145-10   145-10   145-10   146-8    146-8    146-8    147-8    147-8    147-8    148-9
        148-9    148-9    153-11   153-11   153-11   154-12   154-12   154-12   155-17   155-17   155-17   155-17#
SVCINS   87-153#  87-159#  87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
         87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   88-11    88-11
         88-11    88-11    89-12    89-12    90-11    90-11    100-13   100-13   100-13   100-13   100-20   100-20   100-20   100-20
        101-17   101-17   101-21   101-21   101-25   101-25   101-29   101-29   101-39   101-39   101-102  101-102  101-106  101-106
        101-111  101-111  101-115  101-115  101-120  101-120  101-124  101-124  101-128  101-128  101-138  101-138  101-186  101-186
        102-9    102-9    102-9    102-9    102-9    102-9    102-9    102-9    102-11   102-11   102-78   102-78   102-78   102-78
        102-78   102-78   102-78   102-78   102-78   102-78   102-78   102-78   102-80   102-80   102-80   102-80   102-80   102-80
        102-80   102-80   102-80   102-80   102-80   102-80   102-82   102-82   102-82   102-82   102-82   102-82   102-82   102-82
        102-82   102-82   102-82   102-82   102-84   102-84   102-84   102-84   102-84   102-84   102-84   102-84   102-84   102-84
        102-84   102-84   105-26   105-26   105-26   105-26   105-26   105-26   105-26   105-26   105-44   105-44   105-44   105-44
        105-44   105-44   105-44   105-44   105-47   105-47   105-59   105-59   105-94   105-94   105-94   105-94   105-94   105-94
        105-94   105-94   105-112  105-112  105-112  105-112  105-112  105-112  105-112  105-112  114-35   114-35   116-40   116-40
        117-45   117-45   117-56   117-56   121-25   121-25   121-25   121-25   121-25   121-25   121-25   121-25   127-39   127-39
        127-39   127-39   127-39   127-39   127-39   127-39   127-39   127-39   127-43   127-43   127-43   127-43   127-43   127-43
        127-43   127-43   127-43   127-43   128-26   128-26   128-41   128-41   129-20   129-20   129-20   129-20   129-20   129-20
```

Cross reference table (CREF V05.01)

129-20	129-20	129-20	129-20	129-20	129-20	129-91	129-91	129-91	129-91	129-91	129-91	129-91	129-91
133-27	133-27	133-35	133-35	133-35	133-35	133-35	133-35	133-35	133-35	133-48	133-48	133-48	133-48
133-48	133-48	133-48	133-48	134-13	134-13	134-35	134-35	134-54	134-54	136-122	136-122	136-122	136-122
136-122	136-122	136-122	136-122	136-128	136-128	136-128	136-128	136-128	136-128	136-128	136-128	136-134	136-134
136-134	136-134	136-134	136-134	136-134	136-134	137-17	137-17	137-30	137-30	137-30	137-30	137-30	137-30
137-30	137-30	137-30	137-30	137-30	137-30	137-33	137-33	137-33	137-33	137-36	137-36	137-36	137-36
137-36	137-36	137-36	137-36	137-95	137-95	137-95	137-95	137-95	137-95	137-95	137-95	138-30	138-30
138-42	138-42	138-42	138-42	138-42	138-42	138-42	138-42	138-53	138-53	138-53	138-53	138-53	138-53
138-53	138-53	139-14	139-14	139-14	139-14	139-14	139-14	139-14	139-14	139-14	139-14	139-14	139-14
141-26	141-26	141-26	141-26	141-26	141-26	141-26	141-26	141-26	141-26	141-36	141-36	141-36	141-36
141-36	141-36	141-36	141-36	141-36	141-36	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59
141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59
141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61
141-61	141-61	141-82	141-82	141-82	141-82	141-82	141-82	141-82	141-82	141-82	141-82	141-82	141-82
141-82	141-82	141-93	141-93	141-93	141-93	141-107	141-107	143-46	143-46	143-46	143-46	143-48	143-48
143-52	143-52	143-52	143-52	143-54	143-54	143-58	143-58	143-58	143-58	143-60	143-60	143-65	143-65
143-65	143-65	143-67	143-67	143-93	143-93	143-93	143-93	143-95	143-95	143-103	143-103	143-103	143-103
143-103	143-103	143-103	143-103	143-105	143-105	143-127	143-127	143-127	143-127	143-128	143-128	143-129	143-129
143-129	143-129	143-130	143-130	143-140	143-140	143-140	143-140	143-140	143-140	143-140	143-140	143-140	143-140
143-140	143-140	143-146	143-146	143-146	143-146	143-209	143-209	143-209	143-209	143-210	143-210	143-320	143-320
143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320	143-320
143-331	143-331	143-332	143-332	143-333	143-333	143-333	143-333	143-333	143-333	143-333	143-333	143-333	143-333
143-333	143-333	143-337	143-337	143-351	143-351	143-351	143-351	143-353	143-353	143-353	143-353	144-19	144-19
144-19	144-19	144-19	144-19	144-19	144-19	144-21	144-21	144-28	144-28	144-28	144-28	144-28	144-28
144-28	144-28	144-30	144-30	144-42	144-42	145-12	145-12	146-14	146-14	146-14	146-14	146-16	146-16
147-10	147-10	147-10	147-10	147-12	147-12	148-12	148-12	148-12	148-12	148-14	148-14	150-42	150-42
150-43	150-43	150-114	150-114	150-117	150-117	150-118	150-118	150-118	150-118	151-16	151-16	151-16	151-16
151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-18	151-18
151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18
151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26
151-26	151-26	151-33	151-33	151-33	151-33	151-33	151-33	151-33	151-33	151-33	151-33	151-33	151-33
151-39	151-39	151-39	151-39	151-39	151-39	151-39	151-39	151-39	151-39	151-39	151-39	151-45	151-45
151-45	151-45	151-45	151-45	151-45	151-45	151-45	151-45	151-45	151-45	151-46	151-46	151-46	151-46
151-46	151-46	151-46	151-46	151-46	151-46	151-46	151-46	151-49	151-49	151-49	151-49	151-49	151-49
151-49	151-49	151-49	151-49	151-49	151-49	151-54	151-54	151-54	151-54	151-54	151-54	151-54	151-54
151-54	151-54	151-54	151-54	151-54	151-54	151-54	151-54	151-67	151-67	151-67	151-67	151-67	151-67
151-67	151-67	151-67	151-67	151-67	151-67	151-68	151-68	151-68	151-68	151-68	151-68	151-68	151-68
151-68	151-68	151-68	151-68	151-71	151-71	151-71	151-71	151-71	151-71	151-71	151-71	151-71	151-71
151-71	151-71	151-74	151-74	151-74	151-74	151-74	151-74	151-74	151-74	151-74	151-74	151-74	151-74
151-76	151-76	151-76	151-76	151-76	151-76	151-76	151-76	151-76	151-76	151-76	151-76	151-94	151-94
151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-94	151-95	151-95	151-95	151-95	151-95	151-95
151-95	151-95	151-95	151-95	151-96	151-96	151-96	151-96	151-96	151-96	151-96	151-96	151-96	151-96
151-97	151-97	151-97	151-97	151-97	151-97	151-97	151-97	151-97	151-97	151-98	151-98	151-98	151-98
151-98	151-98	151-98	151-98	151-98	151-98	151-99	151-99	151-99	151-99	151-99	151-99	151-99	151-99
151-99	151-99	151-100	151-100	151-100	151-100	151-100	151-100	151-100	151-100	151-100	151-100	151-100	151-100
151-100	151-100	151-100	151-100	151-137	151-137	151-137	151-137	151-137	151-137	151-137	151-137	151-137	151-137
151-137	151-137	151-137	151-137	151-137	151-137	151-143	151-143	151-143	151-143	151-143	151-143	151-143	151-143
151-143	151-143	151-143	151-143	151-143	151-143	151-143	151-143	151-147	151-147	151-147	151-147	151-147	151-147
151-147	151-147	151-147	151-147	151-147	151-147	151-147	151-147	151-147	151-147	151-162	151-162	151-162	151-162
151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-166	151-166
151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166
151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172
151-172	151-172	151-176	151-176	151-176	151-176	151-176	151-176	151-176	151-176	151-176	151-176	151-176	151-176
151-176	151-176	151-176	151-176	151-182	151-182	151-182	151-182	151-182	151-182	151-182	151-182	151-182	151-182
151-182	151-182	151-196	151-196	151-196	151-196	151-196	151-196	151-196	151-196	151-196	151-196	151-196	151-196
151-196	151-196	151-196	151-196	151-200	151-200	151-200	151-200	151-200	151-200	151-200	151-200	151-200	151-200

Cross reference table (CREF V05.01)

	151-200	151-200	151-200	151-200	151-200	151-200	151-213	151-213	151-213	151-213	151-213	151-213	151-213	151-213
	151-213	151-213	151-213	151-213	151-213	151-213	151-213	151-213	151-219	151-219	151-219	151-219	151-219	151-219
	151-219	151-219	151-219	151-219	151-219	151-219	151-219	151-219	151-219	151-219	152-47	152-47	153-11	153-11
	153-27	153-27	153-27	153-27	153-27	153-27	153-27	153-27	153-35	153-35	153-35	153-35	153-35	153-35
	153-35	153-35	153-35	153-35	153-38	153-38	153-38	153-38	153-38	153-38	153-41	153-41	154-12	154-12
	154-37	154-37	154-37	154-37	154-37	154-37	154-39	154-39	154-39	154-39	154-39	154-39	154-39	154-39
	154-39	154-39	154-42	154-42	154-42	154-42	154-42	154-42	154-42	154-42	154-42	154-42	154-44	154-44
	154-44	154-44	154-44	154-44	154-46	154-46	154-46	154-46	154-46	154-46	154-48	154-48	154-48	154-48
	154-48	154-48	154-55	154-55	155-17	155-17	155-17	155-17	155-17	155-17	156-14	156-14	156-14	156-14
SVCSUB	87-1530	87-1610												
SVCTAG	87-1530	87-1630	89-23	89-23	89-23	90-21	90-21	90-21	101-17	101-17	101-17	101-21	101-21	101-21
	101-25	101-25	101-25	101-29	101-29	101-29	101-39	101-39	101-39	101-102	101-104	101-102	101-106	101-106
	101-106	101-111	101-111	101-111	101-115	101-115	101-115	101-120	101-120	101-120	101-124	101-124	101-124	101-128
	101-128	101-128	101-138	101-138	101-138	101-186	101-186	101-186	134-13	134-13	134-13	134-35	134-35	134-35
	134-54	134-54	134-54	141-107	141-107	141-107	143-333	143-333	143-333	144-42	144-42	144-42	145-12	145-12
	145-12	146-16	146-16	146-16	147-12	147-12	147-12	148-14	148-14	148-14	151-18	151-18	151-18	151-26
	151-26	151-26	151-33	151-33	151-33	151-39	151-39	151-39	151-45	151-45	151-45	151-46	151-46	151-46
	151-49	151-49	151-49	151-54	151-54	151-54	151-67	151-67	151-67	151-68	151-68	151-68	151-71	151-71
	151-71	151-74	151-74	151-74	151-76	151-76	151-76	151-100	151-100	151-100	151-137	151-137	151-137	151-143
	151-143	151-143	151-147	151-147	151-147	151-162	151-162	151-162	151-166	151-166	151-166	151-172	151-172	151-172
	151-176	151-176	151-176	151-182	151-182	151-182	151-196	151-196	151-196	151-200	151-200	151-200	151-213	151-213
	151-213	151-219	151-219	151-219	152-47	152-47	152-47	153-41	153-41	153-41	154-55	154-55	154-55	156-14
	156-14	156-14	156-24	156-24	156-24									
SVCTST	87-1530	87-1600	150-3	150-3	150-3									
T$$AU	148-90	148-12	148-14											
T$$AUT	145-100	145-12												
T$$CLE	146-80	146-14	146-16											
T$$DAT	156-14	156-140	156-24											
T$$DU	147-80	147-10	147-12											
T$$MAR	153-11	153-110	153-41											
T$$MW	89-12	89-120	89-23											
T$$INI	143-440	143-353	144-42											
T$$MSG	101-150	101-17	101-190	101-21	101-230	101-25	101-270	101-29	101-370	101-39	101-1000	101-102	101-1040	101-106
	101-1080	101-111	101-1130	101-115	101-1170	101-120	101-1220	101-124	101-1260	101-128	101-1360	101-138	101-1650	101-186
T$$PC	156-130	156-25												
T$$PRO	142-90													
T$$PTA	156-130	156-14	156-140											
T$$RPT	141-100	141-93	141-107											
T$$SOF	154-12	154-120	154-55											
T$$SRV	134-110	134-13	134-320	134-35	134-500	134-54								
T$$SW	90-11	90-110	90-21											
T$$TES	150-30	150-118	152-47											
T$ARGC	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-2020	87-2020	87-2020
	87-2020	87-2020	87-2020	102-78	102-78	102-78	102-780	102-780	102-80	102-80	102-80	102-800	102-800	102-82
	102-82	102-82	102-820	102-820	102-84	102-84	102-84	102-840	102-840	127-39	127-39	127-390	127-43	127-43
	127-430	141-26	141-26	141-260	141-36	141-36	141-360	141-59	141-59	141-59	141-59	141-59	141-59	141-59
	141-59	141-590	141-590	141-590	141-590	141-590	141-590	141-590	141-590	141-61	141-61	141-61	141-61	141-610
	141-610	141-610	141-610	141-82	141-82	141-82	141-82	141-820	141-820	141-820	143-320	143-320	143-320	143-320
	143-320	143-3200	143-3200	143-3200	143-3200	151-16	151-16	151-16	151-16	151-16	151-160	151-160	151-160	151-160
	151-94	151-94	151-940	151-95	151-95	151-950	151-96	151-96	151-960	151-97	151-97	151-970	151-98	151-98
	151-980	151-99	151-99	151-990										
T$CODE	143-333	143-333	143-333	143-3330	143-3330	143-3330	151-18	151-18	151-18	151-180	151-180	151-180	151-26	151-26
	151-26	151-260	151-260	151-260	151-33	151-33	151-33	151-330	151-330	151-330	151-39	151-39	151-39	151-390
	151-390	151-390	151-45	151-45	151-45	151-450	151-450	151-450	151-46	151-46	151-46	151-460	151-460	151-460
	151-49	151-49	151-49	151-490	151-490	151-490	151-54	151-54	151-54	151-540	151-540	151-540	151-67	151-67
	151-67	151-670	151-670	151-670	151-68	151-68	151-68	151-680	151-680	151-680	151-71	151-71	151-71	151-710

Cross reference table (CREF V05.01)

```
        151-71#  151-71#  151-74   151-74   151-74   151-74#  151-74#  151-74#  151-76   151-76   151-76   151-76#  151-76#  151-76#
        151-100  151-100  151-100  151-100#  151-100#  151-100#  151-137  151-137  151-137  151-137#  151-137#  151-137#  151-143  151-143
        151-143  151-143#  151-143#  151-143#  151-147  151-147  151-147  151-147#  151-147#  151-147#  151-162  151-162  151-162  151-162#
        151-162#  151-162#  151-166  151-166  151-166  151-166#  151-166#  151-166#  151-172  151-172  151-172  151-172#  151-172#  151-172#
        151-176  151-176  151-176  151-176#  151-176#  151-176#  151-182  151-182  151-182  151-182#  151-182#  151-182#  151-196  151-196
        151-196  151-196#  151-196#  151-196#  151-200  151-200  151-200  151-200#  151-200#  151-200#  151-213  151-213  151-213  151-213#
        151-213#  151-213#  151-219  151-219  151-219  151-219#  151-219#  151-219#  153-27   153-27   153-27   153-27#  153-27#  153-27#
        153-35   153-35   153-35   153-35#  153-35#  153-35#  153-38   153-38   153-38   153-38#  153-38#  153-38#  154-37   154-37
        154-37   154-37#  154-37#  154-37#  154-39   154-39   154-39   154-39#  154-39#  154-39#  154-42   154-42   154-42   154-42#
        154-42#  154-42#  154-44   154-44   154-44   154-44#  154-44#  154-44#  154-46   154-46   154-46   154-46#  154-46#  154-46#
        154-48   154-48   154-48   154-48#  154-48#  154-48#
T$ERRN   87-153#  102-9    102-9#   105-26   105-26#  105-44   105-44#  105-94   105-94#  105-112  105-112#  121-25   121-25#  129-91
        129-91#  133-35   133-35#  133-48   133-48#  136-122  136-122#  136-128  136-128#  136-134  136-134#  137-36   137-36#  137-95
        137-95#  138-42   138-42#  138-53   138-53#  143-103  143-103#  144-19   144-19#  144-28   144-28#
T$EXCP  151-18   151-18#  151-26   151-26#  151-54   151-54#  151-100  151-100#  151-137  151-137#  151-143  151-143#  151-147  151-147#
        151-162  151-162#  151-166  151-166#  151-172  151-172#  151-176  151-176#  151-196  151-196#  151-200  151-200#  151-213  151-213#
        151-219  151-219#  153-27   153-27#  153-35   153-35#  154-39   154-39#  154-42   154-42#
T$FLAG  141-93   141-93#  141-93#  143-353  143-353  143-353#  143-353#  146-14   146-14   146-14#  146-14#  147-10   147-10#  147-10#
        148-12   148-12#  148-12#  150-118  150-118  150-118#  150-118#
T$FREE  155-17   156-25#
T$GMAN   87-153#  151-18#  151-18#  151-26   151-26#  151-26#  151-54#  151-54#  151-100#  151-100#  151-137#  151-137#  151-143  151-143#
        151-143#  151-147  151-147#  151-147#  151-162#  151-162#  151-166#  151-166#  151-172#  151-172#  151-176#  151-176#  151-196  151-196#
        151-196#  151-200  151-200#  151-200#  151-213#  151-213#  151-219#  151-219#
T$MILI  151-18   151-18#  151-26   151-26#  151-54   151-54#  151-100  151-100#  151-137  151-137#  151-143  151-143#  151-147  151-147#
        151-162  151-162#  151-166  151-166#  151-172  151-172#  151-176  151-176#  151-196  151-196#  151-200  151-200#  151-213  151-213#
        151-219  151-219#  153-27   153-27#  153-35   153-35#  154-39   154-39#  154-42   154-42#
T$LAST   87-153#  155-17#  156-13
T$LOLI  151-18   151-18#  151-26   151-26#  151-54   151-54#  151-100  151-100#  151-137  151-137#  151-143  151-143#  151-147  151-147#
        151-162  151-162#  151-166  151-166#  151-172  151-172#  151-176  151-176#  151-196  151-196#  151-200  151-200#  151-213  151-213#
        151-219  151-219#  153-27   153-27#  153-35   153-35#  154-39   154-39#  154-42   154-42#
T$LSYM   87-153   87-153#  89-23    90-21    101-17   101-21   101-25   101-29   101-39   101-102  101-106  101-111  101-115  101-120
        101-124  101-128  101-138  101-186  134-13   134-35   134-54   141-107  144-42   145-12   146-16   147-12   148-14   152-47
        153-41   154-55
T$LTNO  155-17#
T$NEST   87-153#  89-12    89-12    89-12#   89-23    89-23    89-23    89-23#   90-11    90-11    90-11#   90-21    90-21    90-21
        90-21#   101-15   101-15   101-15#  101-17   101-17   101-17   101-17#  101-19   101-19   101-19#  101-21   101-21   101-21
        101-21#  101-23   101-23   101-23#  101-25   101-25   101-25   101-25#  101-27   101-27   101-27#  101-29   101-29   101-29
        101-29#  101-37   101-37   101-37#  101-39   101-39   101-39   101-39#  101-100  101-100  101-100#  101-102  101-102  101-102
        101-102#  101-104  101-104  101-104#  101-106  101-106  101-106  101-106#  101-108  101-108  101-108#  101-111  101-111  101-111
        101-111#  101-113  101-113  101-113#  101-115  101-115  101-115  101-115#  101-117  101-117  101-117#  101-120  101-120  101-120
        101-120#  101-122  101-122  101-122#  101-124  101-124  101-124  101-124#  101-126  101-126  101-126#  101-128  101-128  101-128
        101-128#  101-136  101-136  101-136#  101-138  101-138  101-138  101-138#  101-165  101-165  101-165#  101-186  101-186  101-186
        101-186#  134-11   134-11   134-11#   134-13   134-13   134-13   134-13#   134-32   134-32   134-32#  134-35   134-35   134-35
        134-35#  134-50   134-50   134-50#  134-54   134-54   134-54   134-54#  141-10   141-10   141-10#  141-107  141-107  141-107
        141-107#  142-9    142-9    142-9#   142-15   142-15   142-15   142-15#  143-44   143-44   143-44#  144-42   144-42   144-42
        144-42#  145-10   145-10   145-10#  145-12   145-12   145-12   145-12#  146-8    146-8    146-8#   146-16   146-16   146-16
        146-16#  147-8    147-8    147-8#   147-12   147-12   147-12   147-12#  148-9    148-9    148-9#   148-14   148-14   148-14
        148-14#  150-3    150-3    150-3#   152-47   152-47   152-47   152-47#  153-11   153-11   153-11#  153-41   153-41   153-41
        153-41#  154-12   154-12   154-12#  154-55   154-55   154-55   154-55#
T$NSO    89-12#   89-23    90-11#   90-21    101-15#  101-17   101-19#  101-21   101-23#  101-25   101-27#  101-29   101-37#  101-39
        101-100#  101-102  101-104#  101-106  101-108#  101-111  101-113#  101-115  101-117#  101-120  101-122#  101-124  101-126#  101-128
        101-136#  101-138  101-165#  101-186  134-11#   134-13   134-32#  134-35   134-50#  134-54   141-10#  141-107  142-9#   142-15
        143-44#  144-42   145-10#  145-12   146-8#   146-16   147-8#   147-12   148-9#   148-14   150-3#   152-47   153-11#  153-41
        154-12#  154-55
T$PCNT  156-13#  156-14   156-14   156-14#
T$PTAB  156-14   156-14#
```

CZUDIAO UDA50-A/KDA50-Q DRIVE E MACRO V05.01b Sunday 07-Oct-84 20:58 Page S-20
Cross reference table (CREF V05.01)

```
T$PTHV   87-202  156-25#
T$PTNU   87-153# 156-14   156-14#  156-25   156-25
T$SAVL   87-153#
T$SEGL   87-153#
T$SIZE  155-17   156-25#
T$SUBN   87-153# 150-3#
T$TAGL   87-153#
T$TAGN   87-153#  89-12    89-12    89-12#   90-11    90-11    90-11#   101-15   101-15   101-15#  101-19   101-19   101-19#  101-23
        101-23   101-23#  101-27   101-27   101-27#  101-37   101-37   101-37#  101-100  101-100  101-100# 101-104  101-104  101-104#
        101-108  101-108  101-108# 101-113  101-113  101-113# 101-117  101-117  101-117# 101-122  101-122  101-122# 101-126  101-126
        101-126# 101-136  101-136  101-136# 101-165  101-165  101-165# 134-11   134-11   134-11#  134-32   134-32   134-32#  134-50
        134-50   134-50#  141-10   141-10   141-10#  142-9    142-9    142-9#   143-44   143-44   143-44#  145-10   145-10   145-10#
        146-8    146-8    146-8#   147-8    147-8    147-8#   148-9    148-9    148-9#   150-3    150-3    150-3#   153-11   153-11
        153-11#  154-12   154-12   154-12#  156-13   156-13   156-13#  156-14   156-14   156-14   156-14   156-14#  156-14#
T$TEMP   88-11    88-11    88-11#   88-11#   89-23    89-23#   90-21    90-21#   101-17   101-17#  101-21   101-21#  101-25   101-25#
        101-29   101-29#  101-39   101-39#  101-102  101-102# 101-106  101-106# 101-111  101-111# 101-115  101-115# 101-120  101-120#
        101-124  101-124# 101-128  101-128# 101-138  101-138# 101-186  101-186# 134-13   134-13#  134-35   134-35#  134-54   134-54#
        141-93   141-93#  141-107  141-107# 142-15   142-15#  143-333  143-333  143-333  143-333# 143-333# 143-333# 143-353  143-353#
        144-42   144-42#  145-12   145-12#  146-14   146-14#  146-16   146-16#  147-10   147-10#  147-12   147-12#  148-12   148-12#
        148-14   148-14#  150-118  150-118# 151-18   151-18   151-18   151-18#  151-18#  151-18#  151-26   151-26   151-26   151-26#
        151-26#  151-26#  151-33   151-33   151-33   151-33#  151-33#  151-33#  151-39   151-39   151-39   151-39#  151-39#  151-39#
        151-45   151-45   151-45   151-45#  151-45#  151-45#  151-46   151-46   151-46   151-46#  151-46#  151-46#  151-49   151-49
        151-49   151-49#  151-49#  151-49#  151-54   151-54   151-54   151-54#  151-54#  151-54#  151-67   151-67   151-67   151-67#
        151-67#  151-67#  151-68   151-68   151-68   151-68#  151-68#  151-68#  151-71   151-71   151-71   151-71#  151-71#  151-71#
        151-74   151-74   151-74   151-74#  151-74#  151-74#  151-76   151-76   151-76   151-76#  151-76#  151-76#  151-100  151-100
        151-100  151-100#  151-100#  151-100#  151-137  151-137  151-137  151-137#  151-137#  151-137#  151-143  151-143  151-143  151-143#
        151-143#  151-143#  151-147  151-147  151-147  151-147#  151-147#  151-147#  151-162  151-162  151-162  151-162#  151-162#  151-162#
        151-166  151-166  151-166  151-166#  151-166#  151-166#  151-172  151-172  151-172  151-172#  151-172#  151-172#  151-176  151-176
        151-176  151-176#  151-176#  151-176#  151-182  151-182  151-182  151-182#  151-182#  151-182#  151-196  151-196  151-196  151-196#
        151-196#  151-196#  151-200  151-200  151-200  151-200#  151-200#  151-200#  151-213  151-213  151-213  151-213#  151-213#  151-213#
        151-219  151-219  151-219  151-219#  151-219#  151-219#  152-47   152-47#  153-27   153-27   153-27   153-27#  153-27#  153-27#
        153-35   153-35   153-35   153-35#  153-35#  153-35#  153-38   153-38   153-38   153-38#  153-38#  153-38#  153-41   153-41#
        154-37   154-37   154-37   154-37#  154-37#  154-37#  154-39   154-39   154-39   154-39#  154-39#  154-39#  154-42   154-42
        154-42   154-42#  154-42#  154-42#  154-44   154-44   154-44   154-44#  154-44#  154-44#  154-46   154-46   154-46   154-46#
        154-46#  154-46#  154-48   154-48   154-48   154-48#  154-48#  154-48#  154-55   154-55#
T$TEST   87-153#  150-3    150-3    150-3#   155-17
T$TSTM   87-153#  101-17   101-21   101-25   101-29   101-39   101-102  101-106  101-111  101-115  101-120  101-124  101-128  101-138
        101-186  102-9    102-11   102-78   102-80   102-82   102-84   105-26   105-44   105-47   105-59   105-94   105-112  114-35
        116-40   117-45   117-56   121-25   127-39   127-43   128-26   128-41   129-20   129-91   133-27   133-35   133-48   136-122
        136-128  136-134  137-17   137-30   137-33   137-36   137-95   138-30   138-42   138-53   139-14   141-26   141-36   141-59
        141-61   141-82   141-107  143-46   143-52   143-58   143-65   143-93   143-103  143-105  143-127  143-129  143-140  143-146
        143-209  143-320  143-331  143-333  143-337  143-351  143-353  144-19   144-21   144-28   144-30   144-42   145-12   146-14
        146-16   147-12   148-14   150-42   150-114  150-117  150-118  151-16   151-18   151-26   151-33   151-39   151-45   151-46
        151-49   151-54   151-67   151-68   151-71   151-74   151-76   151-94   151-95   151-96   151-97   151-98   151-99   151-100
        151-137  151-143  151-147  151-162  151-166  151-172  151-176  151-182  151-196  151-200  151-213  151-219  152-47
T$TSTS   87-153#  150-3#
T1       88-11    150-3#
T4BB    151-18   152-5#
T4BBI   151-26   152-7#
T4BE    151-137  152-18#
T4BEG   151-143  152-19#
T4CON   150-24   150-27   150-29#
T4CYL   151-182  152-25#
T4CYLB  151-196  152-26#
T4CYLE  151-200  152-27#
T4DCA   151-68   152-14#
```

Cross reference table (CREF V05.01)

T4DCR	151-71	152-150			
T4DEF	150-41	150-810			
T4DEFA	150-820	150-98			
T4DEFB	150-850	150-95			
T4DEFC	150-910	150-93			
T4DEFD	150-940				
T4DEFE	150-86	150-960			
T4DEFW	150-43	150-770			
T4DMN	151-33	152-60			
T4DP	151-54	152-120			
T4DPC	151-213	152-280			
T4DPD	151-219	152-290			
T4ECC	151-67	152-130			
T4END	151-147	152-200			
T4EXIT	150-112	150-1160			
T4GRC	151-172	152-230			
T4GRP	151-176	152-240			
T4OPT1	151-94	152-340			
T4OPT2	151-95	152-350			
T4OPT3	151-96	152-360			
T4OPT4	151-97	152-370			
T4OPT5	151-98	152-380			
T4OPT6	151-99	152-390			
T4OPT7	100-260	151-100			
T4PRM1	150-490	150-65			
T4PRM2	150-520	150-62			
T4PRM3	150-55	150-59	150-610		
T4PRM4	150-53	150-630			
T4PRM5	151-2180	151-222			
T4Q01	151-250	151-28	151-30		
T4Q02	151-20	151-310			
T4Q03	151-41	151-440			
T4Q04	151-48	151-510			
T4Q05	151-43	151-50	151-520		
T4Q06	151-610				
T4Q07	151-62	151-64	151-670		
T4Q08	151-70	151-730			
T4Q09	151-66	151-72	151-740		
T4Q10	151-82	151-850			
T4Q11	151-84	151-86	151-89	151-92	151-940
T4Q12	151-102	151-1050			
T4Q13	151-106	151-1120			
T4Q14	151-114	151-1180			
T4Q15	151-121	151-1230			
T4Q16	151-111	151-124	151-1270		
T4Q17	151-1280				
T4Q18	151-1310	151-133			
T4Q19	151-104	151-108	151-117	151-126	151-1340
T4Q20	151-1420	151-145	151-151		
T4Q21	151-1460	151-149			
T4Q22	151-135	151-1530			
T4Q23	151-1650	151-169			
T4Q24	151-160	151-1710			
T4Q25	151-1750	151-179			
T4Q26	151-170	151-1800			
T4Q27	151-156	151-184	151-1900		

Cross reference table (CREF V05.01)

```
T4Q27A 151-191  151-193#
T4Q28  151-195# 151-198
T4Q29  151-199# 151-202
T4Q30  151-36   151-152  151-154  151-189  151-203#
T4QHED 151-16   152-33#
T4QUE  151-211  151-217  151-223#
T4QUE2 150-70   151-209#
T4QUES 150-56   151-15#
T4RET  151-74   152-16#
T4RO   151-39   152-8#
T4RUN  150-39   150-73   150-102#
T4SEK  151-76   152-17#
T4STRT 150-10   150-12   150-30   150-36#
T4TRAK 151-166  152-22#
T4TRC  151-162  152-21#
T4WAIT 150-32   150-109  150-111# 150-115
T4WARN 100-47#  150-77
T4WCA  151-46   152-10#
T4WCR  151-49   152-11#
T4WO   151-45   152-9#
TEMP    99-33#  126-20   127-18   141-39   141-59   141-67   143-333  143-334  144-18   144-27   151-17*  151-18   151-19   151-24
       151-26   151-38*  151-39   151-40   151-42*  151-45   151-46   151-47   151-49   151-51*  151-52   151-53*  151-54   151-55
       151-57   151-60*  151-61   151-63   151-65   151-67   151-68   151-69   151-71   151-73*  151-74   151-75*  151-76   151-77*
       151-78   151-80*  151-83*  151-87*  151-90*  151-93*  151-100  151-101* 151-105* 151-113  151-120* 151-136* 151-137  151-138
       151-139  151-143  151-147  151-161* 151-162  151-163  151-165* 151-166  151-167  151-171* 151-172  151-173  151-175* 151-176
       151-177  151-180* 151-181* 151-182  151-183  151-196  151-200  151-212* 151-213  151-214  151-216  151-218* 151-219  151-220
TEST4  103-34
TINDEX  97-29#  97-30    97-30    97-30#   97-31    97-31    97-31#   97-32    97-32    97-32#   97-33    97-33    97-33#   97-34
        97-34   97-34#   97-35    97-35    97-35#   97-36    97-36    97-36#   97-37    97-37    97-37#   97-38    97-38    97-38#
        97-39   97-39    97-39#   97-40    97-40    97-40#   97-41    97-41    97-41#   97-42    97-42    97-42#   97-43    97-43
        97-43#  97-45    98-33#   98-34    98-34    98-34#   98-35    98-35    98-35#   98-37    98-37    98-37#   98-38    98-38
        98-38#  98-39    98-39    98-39#   98-40    98-40    98-40#   98-41    98-41    98-41#   98-42    98-42    98-42#   98-43
        98-43   98-43#   98-44    98-44    98-44#   98-45    98-45    98-45#   98-46    98-46    98-46#   98-47    98-47    98-47#
        98-48   98-48    98-48#   98-49    98-49    98-49#   98-50    98-50    98-50#   98-51    98-51    98-51#   98-52    98-52
        98-52#  98-53    98-53    98-53#   98-54    98-54    98-54#   98-55    98-55    98-55#   98-56    98-56    98-56#   98-57
        98-57   98-57#   98-58    98-58    98-58#   98-59    98-59    98-59#   98-60    98-60    98-60#   98-61    98-61    98-61#
        98-62   98-62    98-62#   98-63    98-63    98-63#   98-64    98-64    98-64#   98-65    98-65    98-65#   98-66    98-66
        98-66#  98-67    98-67    98-67#   98-68    98-68    98-68#   98-69    98-69    98-69#   98-70    98-70    98-70#   98-71
        98-71   98-71#   98-72    98-72    98-72#   98-73    98-73    98-73#   98-74    98-74    98-74#   98-75    98-75    98-75#
        98-78   153-13#  153-14   153-14   153-14#  153-19   153-19   153-19#  153-21   153-21   153-21#  154-13#  154-15   154-15
       154-15#  154-16   154-16   154-16#  154-18   154-18   154-18#
TINIT  103-15#  150-37
TNAMES 101-172  102-266#
TNUM    99-27#  101-166  103-16*  141-13   143-148* 150-9
TOOMER 143-277  144-27#
TSTTAB  99-17#  104-21   105-10   150-106*
TTYOUT  99-81#  102-68*  102-78   102-80   102-82   102-84
UAM     91-8#
UCNT    99-30#  104-22*  104-33*  105-11*  105-64*  150-47*  150-60*
UF.576  94-109#
UF.CMR  94-100#
UF.CMW  94-101#
UF.INA  94-103#
UF.RPL  94-102#
UF.SCH  94-104#
UF.SCL  94-105#
```

Cross reference table (CREF V05.01)

```
UF.WBN   94-106#
UF.WPH   94-107#
UF.WPS   94-108#
URNING   99-29#   104-17*  104-31*  104-38   105-71*  150-29
URUN     99-28#   104-16*  104-22   105-11
UTOT1   115-21#   115-35
UTOT1A  115-24    115-34#
UTOT2   115-22    115-36#
WAITMS  129-79    133-12#
X$ALWA   87-153#
X$FALS   87-153#
X$OFFS   87-153#
X$TRUE   87-153#
X14     100-100#  101-38
X1A     100-85#
X2      100-92#   101-16
X2A     100-86#   101-16
X3      100-93#   101-20
X31     100-127#  101-105
X32     100-129#  101-109
X35     100-130#  101-118
X36     100-131#  101-123
X38     100-133#  101-127
X3A     100-87#   101-20
X4      100-94#   101-24
X40     100-143#  101-137
X6      100-96#   101-28
X8A     100-88#
XFCE    100-146#  101-101
XFRU    100-154#  120-106
XMSG1   100-147#  101-155
XMSG2   100-148#  101-159
XPKT1   100-149#  101-141
XPKT2   100-152#  101-148
XSA     100-153#  120-118
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
ENDSET    1-5550  87-1530 156-25
ENDSFT    1-5680  87-1530 154-55
ENDSRV    1-5800  87-1530 134-13  134-35  134-54
ENDSUB    1-5960  87-1530
ENDSW     1-6140  87-1530  90-21
ENDTST    1-6240  87-1530 152-47
EQUALS    1-6420  87-1530  91-8
ERRDF     1-7140  87-1530 105-26  105-44  105-94  105-112 121-25  129-91  133-35  133-48  136-122 137-36  137-95  138-42
        138-53
ERRHRD    1-7180  87-1530
ERROR     1-7220  87-1530 128-41
ERRSF     1-7260  87-1530 102-9   136-128 136-134 143-103 144-19  144-28
ERRSOF    1-7300  87-1530
ERRTBL    1-7340  87-1530  99-8
ESCAPE    1-7440  87-1530
EXIT      1-7710  87-1530 141-93  143-353 146-14  147-10  148-12  150-118
FEQUAL    1-8100  87-1530
GETBYT    1-8240  87-1530
GETPRI    1-8340  87-1530
GETWOR    1-8290  87-1530
GMANIA    1-8390  87-1530
GMANID    1-8480  87-1530 151-18  151-26  151-54  151-100 151-137 151-143 151-147 151-162 151-166 151-172 151-176 151-196
        151-200 151-213 151-219
GMANIL    1-8590  87-1530 143-333 151-33  151-39  151-45  151-46  151-49  151-67  151-68  151-71  151-74  151-76  151-182
GPHARD    1-8680  87-1530 143-93  143-209
GPRMA     1-8740  87-1530 153-27
GPRMD     1-9030  87-1530 151-18  151-180 151-26  151-260 151-54  151-540 151-100 151-1000 151-137 151-1370 151-143 151-1430
        151-147 151-1470 151-162 151-1620 151-166 151-1660 151-172 151-1720 151-176 151-1760 151-196 151-1960 151-200 151-2000
        151-213 151-2130 151-219 151-2190 153-35  154-39  154-42
GPRML     1-9340  87-1530 143-333 143-3330 151-33  151-330 151-39  151-390 151-45  151-450 151-46  151-460 151-49  151-490
        151-67  151-670 151-68  151-680 151-71  151-710 151-74  151-740 151-76  151-760 151-182 151-1820 153-38  154-37
        154-44  154-46  154-48
HEADER    1-9540  87-1530  87-202
INLOOP    1-9620  87-1530
IOSETU    1-9660  87-1530
IOSTAR    1-9740  87-1530
ITEM     87-750   97-30   97-31   97-32   97-33   97-34   97-35   97-36   97-37   97-38   97-39   97-40   97-41   97-42
         97-43   98-34   98-35   98-57   98-38   98-39   98-40   98-41   98-42   98-43   98-44   98-45   98-46   98-47
         98-48   98-49   98-50   98-51   98-52   98-53   98-54   98-55   98-56   98-57   98-58   98-59   98-60   98-61
         98-62   98-63   98-64   98-65   98-66   98-67   98-68   98-69   98-70   98-71   98-72   98-73   98-74   98-75
        153-14  153-19  153-21  154-15  154-16  154-18
KT11      1-9820  87-1530
LASTAD    1-;470  87-1530 155-17
M$BYTE    1-D000  87-1530  87-202  87-202  87-202  87-2020
M$CHEC    1-E180  87-1530 141-93  141-930 143-353 143-3530 146-14  146-140 147-10  147-100 148-12  148-120 150-118 150-1180
M$CNTO    1-E820  87-1530 143-333 143-3330 151-18  151-180 151-26  151-260 151-33  151-330 151-39  151-390 151-45  151-450
        151-46  151-460 151-49  151-490 151-54  151-540 151-67  151-670 151-68  151-680 151-71  151-710 151-74  151-740
        151-76  151-760 151-100 151-1000 151-137 151-1370 151-143 151-1430 151-147 151-1470 151-162 151-1620 151-166 151-1660
        151-172 151-1720 151-176 151-1760 151-182 151-1820 151-196 151-1960 151-200 151-2000 151-213 151-2130 151-219 151-2190
        153-27  153-270  153-35  153-350  153-38  153-380  154-37  154-370  154-39  154-390  154-42  154-420  154-44  154-440
        154-46  154-460  154-48  154-480
M$COUN    1-D660  87-1530 102-78  102-780 102-80  102-800 102-82  102-820 102-84  102-840 127-39  127-390 127-43  127-430
        141-26  141-260 141-36  141-360 141-59  141-59  141-59  141-59  141-59  141-59  141-590 141-61  141-61  141-61
        141-610 141-82  141-82  141-820 143-320 143-320 143-320 143-3200 151-16 151-16  151-16  151-160 151-94  151-940
        151-95  151-950 151-96  151-960 151-97  151-970 151-98  151-980 151-99  151-990
M$DATA    1-B670  87-1530  87-202  87-202  87-202  87-202  87-202  87-202  87-202  87-202  87-202  87-202  87-202  87-202
```

Cross reference table (CREF V05.01)

```
        87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
        87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202#  87-202#  100-13
       100-13#  100-20   100-20#
M$DECR   1-D29#  87-153#  89-23    89-23#   90-21    90-21#   101-17   101-17#  101-21   101-21#  101-25   101-25#  101-29   101-29#
       101-39   101-39#  101-102  101-102#  101-106  101-106#  101-111  101-111#  101-115  101-115#  101-120  101-120#  101-124  101-124#
       101-128  101-128#  101-138  101-138#  101-186  101-186#  134-13   134-13#  134-35   134-35#  134-54   134-54#  141-107  141-107#
       142-15   142-15#  144-42   144-42#  145-12   145-12#  146-16   146-16#  147-12   147-12#  148-14   148-14#  152-47   152-47#
       153-41   153-41#  154-55   154-55#  156-14   156-14#
M$DEFA   1-E70#  87-153#  143-333  143-333#  151-18   151-18#  151-26   151-26#  151-33   151-33#  151-39   151-39#  151-45   151-45#
       151-46   151-46#  151-49   151-49#  151-54   151-54#  151-67   151-67#  151-68   151-68#  151-71   151-71#  151-74   151-74#
       151-76   151-76#  151-100  151-100#  151-137  151-137#  151-143  151-143#  151-147  151-147#  151-162  151-162#  151-166  151-166#
       151-172  151-172#  151-176  151-176#  151-182  151-182#  151-196  151-196#  151-200  151-200#  151-213  151-213#  151-219  151-219#
       153-27   153-27#  153-35   153-35#  153-38   153-38#  154-37   154-37#  154-39   154-39#  154-42   154-42#  154-44   154-44#
       154-46   154-46#  154-48   154-48#
M$ENDE   1-D74#  87-153#  89-23#   90-21#   101-17#  101-21#  101-25#  101-29#  101-39#  101-102#  101-106#  101-111#  101-115#  101-120#
       101-124#  101-128#  101-138#  101-186#  134-13#  134-35#  134-54#  141-107#  144-42#  145-12#  146-16#  147-12#  148-14#  152-47#
       153-41#  154-55#
M$ERRI   1-B49#  87-153#  102-9    102-9#   105-26   105-26#  105-44   105-44#  105-94   105-94#  105-112  105-112#  121-25   121-25#
       129-91   129-91#  133-35   133-35#  133-48   133-48#  136-122  136-122#  136-128  136-128#  136-134  136-134#  137-36   137-36#
       137-95   137-95#  138-42   138-42#  138-53   138-53#  143-103  143-103#  144-19   144-19#  144-28   144-28#
M$ESCA   1-D06#  87-153#
M$ESCS   1-D10#  87-153#
M$EXCP   1-E01#  87-153#  151-18   151-18   151-18#  151-26   151-26   151-26#  151-54   151-54   151-54#  151-100  151-100  151-100#
       151-137  151-137  151-137#  151-143  151-143  151-143#  151-147  151-147  151-147#  151-162  151-162  151-162#  151-166  151-166
       151-166#  151-172  151-172  151-172#  151-176  151-176  151-176#  151-196  151-196  151-196#  151-200  151-200  151-200#  151-213
       151-213  151-213#  151-219  151-219  151-219#  153-27   153-27   153-27#  153-35   153-35   153-35#  154-39   154-39   154-39#
       154-42   154-42   154-42#
M$EXIT   1-D14#  87-153#  141-93#  143-353  143-353#  146-14   146-14#  147-10#  148-12#  150-118  150-118#
M$EXSE   1-D22#  87-153#  141-93#  143-353#  146-14#  147-10#  148-12#  150-118#
M$EXTJ   1-D18#  87-153#  141-93   141-93#  143-353#  146-14#  147-10   147-10#  148-12   148-12#  150-118#
M$GEN    1-D38#  87-153#  87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
        87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202
        87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202   87-202#  87-202#
        87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#
        87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#
        87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  87-202#  88-11   88-11#  89-12   89-12   89-12#  89-12#
        89-23    89-23#   90-11    90-11    90-11#   90-11#   90-21    90-21#   99-8    99-8#   100-13   100-13#  100-20   100-20#
       101-15   101-15#  101-17   101-17#  101-19   101-19#  101-21   101-21#  101-23   101-23#  101-25   101-25#  101-27   101-27#
       101-29   101-29#  101-37   101-37#  101-39   101-39#  101-100  101-100#  101-102  101-102#  101-104  101-104#  101-106  101-106#
       101-108  101-108#  101-111  101-111#  101-113  101-113#  101-115  101-115#  101-117  101-117#  101-120  101-120#  101-122  101-122#
       101-124  101-124#  101-126  101-126#  101-128  101-128#  101-136  101-136#  101-138  101-138#  101-165  101-165#  101-186  101-186#
       134-11   134-11#  134-13   134-13#  134-32   134-32#  134-35   134-35#  134-50   134-50#  134-54   134-54#  141-10   141-10#
       141-107  141-107#  142-9    142-9#   143-44   143-44#  143-333  143-333#  144-42   144-42#  145-10   145-10#  145-12   145-12#
       146-8    146-8#   146-16   146-16#  147-8    147-8#   147-12   147-12#  148-9    148-9#   148-14   148-14#  150-3    150-3#
       151-18   151-18#  151-26   151-26#  151-33   151-33#  151-39   151-39#  151-45   151-45#  151-46   151-46#  151-49   151-49#
       151-54   151-54#  151-67   151-67#  151-68   151-68#  151-71   151-71#  151-74   151-74#  151-76   151-76#  151-100  151-100#
       151-137  151-137#  151-143  151-143#  151-147  151-147#  151-162  151-162#  151-166  151-166#  151-172  151-172#  151-176  151-176#
       151-182  151-182#  151-196  151-196#  151-200  151-200#  151-213  151-213#  151-219  151-219#  152-47   152-47#  153-11   153-11#
       153-41   153-41#  154-12   154-12#  154-55   154-55#  155-17   155-17#  156-14   156-14#  156-24   156-24#
M$GENB   1-C38#  87-153#  143-333  143-333#  151-18   151-18#  151-26   151-26#  151-33   151-33#  151-39   151-39#  151-45   151-45#
       151-46   151-46#  151-49   151-49#  151-54   151-54#  151-67   151-67#  151-68   151-68#  151-71   151-71#  151-74   151-74#
       151-76   151-76#  151-100  151-100#  151-137  151-137#  151-143  151-143#  151-147  151-147#  151-162  151-162#  151-166  151-166#
       151-172  151-172#  151-176  151-176#  151-182  151-182#  151-196  151-196#  151-200  151-200#  151-213  151-213#  151-219  151-219#
M$GETS   1-D35#  87-153#  89-23    89-23#   90-21    90-21#   101-17   101-17#  101-21   101-21#  101-25   101-25#  101-29   101-29#
       101-39   101-39#  101-102  101-102#  101-106  101-106#  101-111  101-111#  101-115  101-115#  101-120  101-120#  101-124  101-124#
       101-128  101-128#  101-138  101-138#  101-186  101-186#  134-13   134-13#  134-35   134-35#  134-54   134-54#  141-107  141-107#
```

Cross reference table (CREF V05.01)

	142-15	142-15#	144-42	144-42#	145-12	145-12#	146-16	146-16#	147-12	147-12#	148-14	148-14#	152-47	152-47#
	153-41	153-41#	154-55	154-55#										
M$GETT	1-877#	87-153#	141-93#	143-353#	146-14#	147-10#	148-12#	150-118#						
M$GNGB	1-C02#	87-153#	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202
	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202
	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202#	87-202#
	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#
	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#
	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	88-11	88-11#	89-12	89-12	89-12#	90-11
	90-11	90-11#	99-8	99-8#	100-13	100-13#	100-20	100-20#	101-15	101-15#	101-19	101-19#	101-23	101-23#
	101-27	101-27#	101-37	101-37#	101-100	101-100#	101-104	101-104#	101-108	101-108#	101-113	101-113#	101-117	101-117#
	101-122	101-122#	101-126	101-126#	101-136	101-136#	101-165	101-165#	134-11	134-11#	134-32	134-32#	134-50	134-50#
	141-10	141-10#	142-9	142-9#	143-44	143-44#	145-10	145-10#	146-8	146-8#	147-8	147-8#	148-9	148-9#
	153-11	153-11#	154-12	154-12#	155-17	155-17#								
M$GNIN	1-D49#	87-153#	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202
	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202
	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202
	87-202	87-202	87-202	87-202	87-202	87-202	87-202	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#
	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#
	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#
	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	87-202#	88-11	88-11	88-11#	88-11#	89-12	89-12#
	90-11	90-11#	100-13	100-13	100-13#	100-13#	100-20	100-20	100-20#	100-20#	101-17	101-17#	101-21	101-21#
	101-25	101-25#	101-29	101-29#	101-39	101-39#	101-102	101-102#	101-106	101-106#	101-111	101-111#	101-115	101-115#
	101-120	101-120#	101-124	101-124#	101-128	101-128#	101-138	101-138#	101-186	101-186#	102-9	102-9	102-9	102-9
	102-9#	102-9#	102-9#	102-9#	102-9#	102-11	102-11#	102-78	102-78	102-78	102-78	102-78	102-78	102-78#
	102-78#	102-78#	102-78#	102-78#	102-80	102-80	102-80	102-80	102-80	102-80	102-80#	102-80#	102-80#	102-80#
	102-80#	102-82	102-82	102-82	102-82	102-82	102-82	102-82#	102-82#	102-82#	102-82#	102-82#	102-84	102-84
	102-84	102-84	102-84	102-84	102-84#	102-84#	102-84#	102-84#	102-84#	105-26	105-26	105-26	105-26	105-26#
	105-26#	105-26#	105-26#	105-26#	105-44	105-44	105-44	105-44	105-44#	105-44#	105-44#	105-44#	105-44#	105-47
	105-47#	105-59	105-59#	105-94	105-94	105-94	105-94	105-94#	105-94#	105-94#	105-94#	105-94#	105-112	105-112
	105-112	105-112	105-112#	105-112#	105-112#	105-112#	105-112#	114-35	114-35#	114-35#	116-40	116-40#	116-40#	117-45
	117-45#	117-45#	117-56	117-56#	121-25	121-25	121-25	121-25	121-25#	121-25#	121-25#	121-25#	121-25#	127-39
	127-39	127-39	127-39	127-39	127-39#	127-39#	127-39#	127-39#	127-43	127-43	127-43	127-43	127-43	127-43#
	127-43#	127-43#	127-43#	128-26	128-26#	128-41	128-41#	129-20	129-20	129-20	129-20	129-20	129-20	129-20#
	129-20#	129-20#	129-20#	129-20#	129-20#	129-91	129-91	129-91	129-91	129-91#	129-91#	129-91#	129-91#	129-91#
	133-27	133-27#	133-35	133-35	133-35	133-35	133-35#	133-35#	133-35#	133-35#	133-35#	133-48	133-48	133-48
	133-48	133-48#	133-48#	133-48#	133-48#	133-48#	134-13	134-13#	134-35	134-35#	134-54	134-54#	136-122	136-122
	136-122	136-122	136-122#	136-122#	136-122#	136-122#	136-122#	136-128	136-128	136-128	136-128	136-128#	136-128#	136-128#
	136-128#	136-128#	136-134	136-134	136-134	136-134	136-134#	136-134#	136-134#	136-134#	136-134#	137-17	137-17#	137-30
	137-30	137-30	137-30	137-30	137-30	137-30#	137-30#	137-30#	137-30#	137-30#	137-30#	137-33	137-33	137-33#
	137-33#	137-36	137-36	137-36	137-36	137-36#	137-36#	137-36#	137-36#	137-36#	137-95	137-95	137-95	137-95
	137-95#	137-95#	137-95#	137-95#	137-95#	138-30	138-30#	138-42	138-42	138-42	138-42	138-42#	138-42#	138-42#
	138-42#	138-42#	138-53	138-53	138-53	138-53	138-53#	138-53#	138-53#	138-53#	138-53#	139-14	139-14	139-14
	139-14	139-14	139-14	139-14#	139-14#	139-14#	139-14#	139-14#	139-14#	139-14#	141-26	141-26	141-26	141-26
	141-26#	141-26#	141-26#	141-26#	141-36	141-36	141-36	141-36	141-36	141-36#	141-36#	141-36#	141-36#	141-59
	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59#	141-59#	141-59#	141-59#
	141-59#	141-59#	141-59#	141-59#	141-59#	141-59#	141-61	141-61	141-61	141-61	141-61	141-61	141-61	141-61
	141-61#	141-61#	141-61#	141-61#	141-61#	141-61#	141-61#	141-82	141-82	141-82	141-82	141-82	141-82	141-82
	141-82#	141-82#	141-82#	141-82#	141-82#	141-82#	141-93	141-93	141-93#	141-93#	141-107	141-107#	143-46	143-46
	143-46#	143-46#	143-48	143-48#	143-52	143-52	143-52#	143-52#	143-54	143-54#	143-58	143-58	143-58#	143-58#
	143-60	143-60#	143-65	143-65	143-65#	143-65#	143-67	143-67#	143-93	143-93	143-93#	143-93#	143-93#	143-95
	143-95#	143-103	143-103	143-103	143-103	143-103#	143-103#	143-103#	143-103#	143-103#	143-105	143-105#	143-127	143-127
	143-127#	143-127#	143-127#	143-128	143-128#	143-129	143-129	143-129#	143-129#	143-129#	143-130	143-130#	143-140	143-140
	143-140	143-140	143-140	143-140	143-140#	143-140#	143-140#	143-140#	143-140#	143-140#	143-146	143-146	143-146#	143-146#
	143-209	143-209	143-209#	143-209#	143-209#	143-210	143-210#	143-320	143-320	143-320	143-320	143-320	143-320	143-320
	143-320	143-320#	143-320#	143-320#	143-320#	143-320#	143-320#	143-320#	143-331	143-331#	143-332	143-332#	143-333	143-333

Cross reference table (CREF V05.01)

	143-333	143-333	143-333	143-333	143-333#	143-333#	143-333#	143-333#	143-337	143-337#	143-351	143-351	143-351#	143-351#
	143-353	143-353	143-353#	143-353#	144-19	144-19	144-19	144-19	144-19#	144-19#	144-19#	144-19#	144-19#	144-21
	144-21#	144-28	144-28	144-28	144-28	144-28#	144-28#	144-28#	144-28#	144-28#	144-30	144-30#	144-42	144-42#
	145-12	145-12#	146-14	146-14	146-14#	146-14#	146-16	146-16#	147-10	147-10	147-10#	147-10#	147-12	147-12#
	148-12	148-12	148-12#	148-12#	148-14	148-14#	150-42	150-42#	150-43	150-43#	150-114	150-114#	150-117	150-117#
	150-118	150-118	150-118#	150-118#	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16	151-16#	151-16#
	151-16#	151-16#	151-16#	151-16#	151-16#	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18	151-18#
	151-18#	151-18#	151-18#	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26	151-26#	151-26#	151-26#
	151-26#	151-33	151-33	151-33	151-33	151-33	151-33	151-33#	151-33#	151-33#	151-33#	151-39	151-39	151-39
	151-39	151-39	151-39	151-39#	151-39#	151-39#	151-39#	151-45	151-45	151-45	151-45	151-45	151-45	151-45#
	151-45#	151-45#	151-45#	151-46	151-46	151-46	151-46	151-46	151-46	151-46#	151-46#	151-46#	151-46#	151-49
	151-49	151-49	151-49	151-49	151-49	151-49#	151-49#	151-49#	151-49#	151-54	151-54	151-54	151-54	151-54
	151-54	151-54	151-54	151-54#	151-54#	151-54#	151-54#	151-67	151-67	151-67	151-67	151-67	151-67	151-67#
	151-67#	151-67#	151-67#	151-68	151-68	151-68	151-68	151-68	151-68	151-68#	151-68#	151-68#	151-68#	151-71
	151-71	151-71	151-71	151-71	151-71	151-71#	151-71#	151-71#	151-71#	151-74	151-74	151-74	151-74	151-74
	151-74	151-74#	151-74#	151-74#	151-74#	151-76	151-76	151-76	151-76	151-76	151-76	151-76#	151-76#	151-76#
	151-76#	151-94	151-94	151-94	151-94	151-94	151-94#	151-94#	151-94#	151-94#	151-95	151-95	151-95	151-95
	151-95	151-95#	151-95#	151-95#	151-95#	151-96	151-96	151-96	151-96	151-96	151-96#	151-96#	151-96#	151-96#
	151-97	151-97	151-97	151-97	151-97	151-97#	151-97#	151-97#	151-97#	151-98	151-98	151-98	151-98	151-98
	151-98#	151-98#	151-98#	151-98#	151-99	151-99	151-99	151-99	151-99	151-99#	151-99#	151-99#	151-99#	151-100
	151-100	151-100	151-100	151-100	151-100	151-100	151-100	151-100#	151-100#	151-100#	151-100#	151-137	151-137	151-137
	151-137	151-137	151-137	151-137	151-137	151-137#	151-137#	151-137#	151-137#	151-143	151-143	151-143	151-143	151-143
	151-143	151-143	151-143	151-143#	151-143#	151-143#	151-143#	151-147	151-147	151-147	151-147	151-147	151-147	151-147
	151-147	151-147#	151-147#	151-147#	151-147#	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162	151-162#
	151-162#	151-162#	151-162#	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166	151-166#	151-166#	151-166#
	151-166#	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172	151-172#	151-172#	151-172#	151-172#	151-176
	151-176	151-176	151-176	151-176	151-176	151-176	151-176	151-176#	151-176#	151-176#	151-176#	151-182	151-182	151-182
	151-182	151-182	151-182	151-182#	151-182#	151-182#	151-182#	151-196	151-196	151-196	151-196	151-196	151-196	151-196
	151-196	151-196#	151-196#	151-196#	151-196#	151-200	151-200	151-200	151-200	151-200	151-200	151-200	151-200	151-200#
	151-200#	151-200#	151-200#	151-213	151-213	151-213	151-213	151-213	151-213	151-213	151-213	151-213#	151-213#	151-213#
	151-213#	151-219	151-219	151-219	151-219	151-219	151-219	151-219	151-219	151-219#	151-219#	151-219#	151-219#	152-47
	152-47#	153-11	153-11#	153-27	153-27	153-27	153-27	153-27#	153-35	153-35	153-35	153-35	153-35	153-35#
	153-38	153-38	153-38	153-38#	153-41	153-41#	154-12	154-12#	154-37	154-37	154-37	154-37#	154-39	154-39
	154-39	154-39	154-39	154-39#	154-42	154-42	154-42	154-42	154-42	154-42#	154-44	154-44	154-44	154-44#
	154-46	154-46	154-46	154-46#	154-48	154-48	154-48	154-48#	154-55	154-55#	155-17	155-17	155-17	155-17#
	156-14	156-14	156-14#	156-14#										
M$GNLS	1-C13#	87-153#	143-333	143-333#	151-18	151-18#	151-26	151-26#	151-33	151-33#	151-39	151-39#	151-45	151-45#
	151-46	151-46#	151-49	151-49#	151-54	151-54#	151-67	151-67#	151-68	151-68#	151-71	151-71#	151-74	151-74#
	151-76	151-76#	151-100	151-100#	151-137	151-137#	151-143	151-143#	151-147	151-147#	151-162	151-162#	151-166	151-166#
	151-172	151-172#	151-176	151-176#	151-182	151-182#	151-196	151-196#	151-200	151-200#	151-213	151-213#	151-219	151-219#
M$GNSU	1-B98#	87-153#												
M$GNTA	1-B90#	87-153#	89-23	89-23#	90-21	90-21#	101-17	101-17#	101-21	101-21#	101-25	101-25#	101-29	101-29#
	101-39	101-39#	101-102	101-102#	101-106	101-106#	101-111	101-111#	101-115	101-115#	101-120	101-120#	101-124	101-124#
	101-128	101-128#	101-138	101-138#	101-186	101-186#	134-13	134-13#	134-35	134-35#	134-54	134-54#	141-107	141-107#
	144-42	144-42#	145-12	145-12#	146-16	146-16#	147-12	147-12#	148-14	148-14#	152-47	152-47#	153-41	153-41#
	154-55	154-55#	156-14	156-14#	156-24	156-24#								
M$GNTE	1-B94#	87-153#	150-3	150-3#										
M$HAPT	1-A39#	87-153#	87-202	87-202#										
M$HNAP	1-B24#	87-153#	87-202	87-202#										
M$INCR	1-D26#	87-153#	89-12	89-12	89-12#	89-12#	90-11	90-11	90-11#	90-11#	101-15	101-15	101-15#	101-15#
	101-17#	101-19	101-19	101-19#	101-19#	101-21#	101-23	101-23	101-23#	101-23#	101-25#	101-27	101-27	101-27#
	101-27#	101-29#	101-37	101-37	101-37#	101-37#	101-39#	101-100	101-100	101-100#	101-100#	101-102#	101-104	101-104
	101-104#	101-104#	101-106#	101-108	101-108	101-108#	101-108#	101-111#	101-113	101-113	101-113#	101-113#	101-115#	101-117
	101-117	101-117#	101-117#	101-120#	101-122	101-122	101-122#	101-122#	101-124#	101-126	101-126	101-126#	101-126#	101-128#
	101-136	101-136	101-136#	101-136#	101-138#	101-165	101-165	101-165#	101-165#	101-186#	102-9#	102-11#	102-78#	102-80#
	102-82#	102-84#	105-26#	105-44#	105-47#	105-59#	105-94#	105-112#	114-35#	116-40#	117-45#	117-56#	121-25#	127-39#

Cross reference table (CREF V05.01)

	127-43#	128-26#	128-41#	129-20#	129-91#	133-27#	133-35#	133-48#	134-11	134-11	134-11#	134-11#	134-32	134-32
	134-32#	134-32#	134-50	134-50	134-50#	134-50#	136-122#	136-128#	136-134#	137-17#	137-30#	137-33#	137-36#	137-95#
	138-30#	138-42#	138-53#	139-14#	141-10	141-10	141-10#	141-10#	141-26#	141-36#	141-59#	141-61#	141-82#	141-107#
	142-9	142-9	142-9#	142-9#	143-44	143-44	143-44#	143-44#	143-46#	143-52#	143-58#	143-65#	143-93#	143-103#
	143-105#	143-127#	143-129#	143-140#	143-146#	143-209#	143-320#	143-331#	143-333	143-333#	143-333#	143-337#	143-351#	143-353#
	144-19#	144-21#	144-28#	144-30#	144-42#	145-10	145-10	145-10#	145-10#	145-12#	146-8	146-8	146-8#	146-8#
	146-14#	146-16#	147-8	147-8	147-8#	147-8#	147-12#	148-9	148-9	148-9#	148-9#	148-14#	150-3	150-3
	150-3	150-3#	150-3#	150-3#	150-42#	150-114#	150-117#	150-118#	151-16#	151-18	151-18#	151-18#	151-26	151-26#
	151-26#	151-33	151-33#	151-3?#	151-39	151-39#	151-39#	151-45	151-45#	151-45#	151-46	151-46#	151-46#	151-49
	151-49#	151-49#	151-54	151-?4#	151-54#	151-67	151-67#	151-67#	151-68	151-68#	151-68#	151-71	151-71#	151-71#
	151-74	151-74#	151-74#	151-76	151-76#	151-76#	151-94#	151-95#	151-96#	151-97#	151-98#	151-99#	151-100	151-100#
	151-100#	151-137	151-137#	151-137#	151-143	151-143#	151-143#	151-147	151-147#	151-147#	151-162	151-162#	151-162#	151-166
	151-166#	151-166#	151-172	151-172#	151-172#	151-176	151-176#	151-176#	151-182	151-182#	151-182#	151-196	151-196#	151-196#
	151-200	151-200#	151-200#	151-213	151-213#	151-213#	151-219	151-219#	151-219#	152-47#	153-11	153-11	153-11#	153-11#
	154-12	154-12	154-12#	154-12#	156-13	156-13#	156-14	156-14	156-14	156-14#				
M$IOSE	1-A00#	87-153#												
M$LDRO	1-C42#	87-153#	137-33	137-33#	143-46	143-46#	143-52	143-52#	143-58	143-58#	143-65	143-65#	143-93	143-93#
	143-127	143-127#	143-129	143-129#	143-209	143-209#	143-351	143-351#						
M$MASK	1-871#	87-153#												
M$MCHI	1-4#	87-153	87-153#	87-153#										
M$MCLO	1-824#	87-153	87-153#	87-153#										
M$MSK1	1-877#	87-153#												
M$POP	1-881#	87-153#	89-23	89-23#	90-21	90-21#	101-17	101-17#	101-21	101-21#	101-25	101-25#	101-29	101-29#
	101-39	101-39#	101-102	101-102#	101-106	101-106#	101-111	101-111#	101-115	101-115#	101-120	101-120#	101-124	101-124#
	101-128	101-128#	101-138	101-138#	101-186	101-186#	134-13	134-13#	134-35	134-35#	134-54	134-54#	141-107	141-107#
	142-15	142-15#	144-42	144-42#	145-12	145-12#	146-16	146-16#	147-12	147-12#	148-14	148-14#	152-47	152-47#
	153-41	153-41#	154-55	154-55#										
M$PRIN	1-836#	87-153#	102-78	102-78#	102-80	102-80#	102-82	102-82#	102-84	102-84#	127-39	127-39#	127-43	127-43#
	141-26	141-26#	141-36	141-36#	141-59	141-59#	141-61	141-61#	141-82	141-82#	143-320	143-320#	151-16	151-16#
	151-94	151-94#	151-95	151-95#	151-96	151-96#	151-97	151-97#	151-98	151-98#	151-99	151-99#		
M$PUSH	1-831#	87-153#	89-12	89-12#	90-11	90-11#	101-15	101-15#	101-19	101-19#	101-23	101-23#	101-27	101-27#
	101-37	101-37#	101-100	101-100#	101-104	101-104#	101-108	101-108#	101-113	101-113#	101-117	101-117#	101-122	101-122#
	101-126	101-126#	101-136	101-136#	101-165	101-165#	134-11	134-11#	134-32	134-32#	134-50	134-50#	141-10	141-10#
	142-9	142-9#	143-44	143-44#	145-10	145-10#	146-8	146-8#	147-8	147-8#	148-9	148-9#	150-3	150-3#
	153-11	153-11#	154-12	154-12#										
M$PUT	1-C72#	87-153#	102-78	102-78	102-78	102-78#	102-80	102-80	102-80	102-80#	102-82	102-82	102-82	102-82#
	102-84	102-84	102-84	102-84#	127-39	127-39	127-39#	127-43	127-43	127-43#	129-20	129-20	129-20	129-20
	129-20#	137-30	137-30	137-30	137-30	137-30#	139-14	139-14	139-14	139-14	139-14#	141-26	141-26	141-26#
	141-36	141-36	141-36#	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59	141-59#	141-61	141-61
	141-61	141-61	141-61	141-61#	141-82	141-82	141-82	141-82	141-82#	143-140	143-140	143-140	143-140	143-140#
	143-320	143-320	143-320	143-320	143-320	143-320#	151-16	151-16	151-16	151-16	151-16	151-16#	151-94	151-94
	151-94#	151-95	151-95	151-95#	151-96	151-9?	151-96#	151-97	151-97	151-97#	151-98	151-98	151-98#	151-99
	151-99	151-99#												
M$PUT1	1-C81#	87-153#	102-78	102-78	102-78	102-78#	102-78#	102-78#	102-80	102-80	102-80	102-80#	102-80#	102-80#
	102-82	102-82	102-82	102-82#	102-82#	102-82#	102-84	102-84	102-84	102-84#	102-84#	102-84#	127-39	127-39
	127-39#	127-39#	127-43	127-43	127-43#	127-43#	129-20	129-20	129-20	129-20	129-20#	129-20#	129-20#	129-20#
	137-30	137-30	137-30	137-30	137-30#	137-30#	137-30#	137-30#	139-14	139-14	139-14	139-14	139-14#	139-14#
	139-14#	139-14#	141-26	141-26	141-26#	141-26#	141-36	141-36	141-36#	141-36#	141-59	141-59	141-59	141-59
	141-59	141-59	141-59	141-59	141-59#	141-59#	141-59#	141-59#	141-59#	141-59#	141-59#	141-59#	141-61	141-61
	141-61	141-61	141-61	141-61#	141-61#	141-61#	141-61#	141-61#	141-82	141-82	141-82	141-82	141-82#	141-82#
	141-82#	141-82#	143-140	143-140	143-140	143-140	143-140#	143-140#	143-140#	143-140#	143-320	143-320	143-320	143-320
	143-320	143-320#	143-320#	143-320#	143-320#	143-320#	151-16	151-16	151-16	151-16	151-16	151-16#	151-16#	151-16#
	151-16#	151-16#	151-94	151-94	151-94#	151-94#	151-95	151-95	151-95#	151-95#	151-96	151-96	151-96#	151-96#
	151-97	151-97	151-97#	151-97#	151-98	151-98	151-98#	151-98#	151-99	151-99	151-99#	151-99#		
M$RADI	1-D77#	87-153#	143-333	143-333#	151-18	151-18#	151-26	151-26#	151-33	151-33#	151-39	151-39#	151-45	151-45#
	151-46	151-46#	151-49	151-49#	151-54	151-54#	151-67	151-67#	151-68	151-68#	151-71	151-71#	151-74	151-74#

Cross reference table (CREF V05.01)

	151-76	151-76#	151-100	151-100#	151-137	151-137#	151-143	151-143#	151-147	151-147#	151-162	151-162#	151-166	151-166#
	151-172	151-172#	151-176	151-176#	151-182	151-182#	151-196	151-196#	151-200	151-200#	151-213	151-213#	151-219	151-219#
	153-27	153-27#	153-35	153-35#	153-38	153-38#	154-37	154-37#	154-39	154-39#	154-42	154-42#	154-44	154-44#
	154-46	154-46#	154-48	154-48#										
M$RBRO	1-C52#	87-153#												
M$RNRO	1-C62#	87-153#	114-35	114-35#	116-40	116-40#	117-45	117-45#	143-93	143-93#	143-127	143-127#	143-129	143-129#
	143-146	143-146#	143-209	143-209#										
M$SETS	1-D32#	87-153#	89-12	89-12#	90-11	90-11#	101-15	101-15#	101-19	101-19#	101-23	101-23#	101-27	101-27#
	101-37	101-37#	101-100	101-100#	101-104	101-104#	101-108	101-108#	101-113	101-113#	101-117	101-117#	101-122	101-122#
	101-126	101-126#	101-136	101-136#	101-165	101-165#	134-11	134-11#	134-32	134-32#	134-50	134-50#	141-10	141-10#
	142-9	142-9#	143-44	143-44#	145-10	145-10#	146-8	146-8#	147-8	147-8#	148-9	148-9#	150-3	150-3#
	153-11	153-11#	154-12	154-12#										
M$STAR	1-A33#	87-153#												
M$SVC	1-C33#	87-153#	101-17	101-17#	101-21	101-21#	101-25	101-25#	101-29	101-29#	101-39	101-39#	101-102	101-102#
	101-106	101-106#	101-111	101-111#	101-115	101-115#	101-120	101-120#	101-124	101-124#	101-128	101-128#	101-138	101-138#
	101-186	101-186#	102-9	102-11	102-11#	102-78	102-78#	102-80	102-80#	102-82	102-82#	102-84	102-84#	105-26
	105-44	105-47	105-47#	105-59	105-59#	105-94	105-112	114-35	114-35#	116-40	116-40#	117-45	117-45#	117-56
	117-56#	121-25	127-39	127-39#	127-43	127-43#	128-26	128-26#	128-41	128-41#	129-20	129-20#	129-91	133-27
	133-27#	133-35	133-48	136-122	136-128	136-134	137-17	137-17#	137-30	137-30#	137-33	137-33#	137-36	137-95
	138-30	138-30#	138-42	138-53	139-14	139-14#	141-26	141-26#	141-36	141-36#	141-59	141-59#	141-61	141-61#
	141-82	141-82#	141-93#	141-107	141-107#	143-46	143-46#	143-52	143-52#	143-58	143-58#	143-65	143-65#	143-93
	143-93#	143-103	143-105	143-105#	143-127	143-127#	143-129	143-129#	143-140	143-140#	143-146	143-146#	143-209	143-209#
	143-320	143-320#	143-331	143-331#	143-333	143-333#	143-337	143-337#	143-351	143-351#	143-353	143-353#	144-19	144-21
	144-21#	144-28	144-30	144-30#	144-42	144-42#	145-12	145-12#	146-14	146-14#	146-16	146-16#	147-10#	147-12
	147-12#	148-12#	148-14	148-14#	150-42	150-42#	150-114	150-114#	150-117	150-117#	150-118	150-118#	151-16	151-16#
	151-18	151-18#	151-26	151-26#	151-33	151-33#	151-39	151-39#	151-45	151-45#	151-46	151-46#	151-49	151-49#
	151-54	151-54#	151-67	151-67#	151-68	151-68#	151-71	151-71#	151-74	151-74#	151-76	151-76#	151-94	151-94#
	151-95	151-95#	151-96	151-96#	151-97	151-97#	151-98	151-98#	151-99	151-99#	151-100	151-100#	151-137	151-137#
	151-143	151-143#	151-147	151-147#	151-162	151-162#	151-166	151-166#	151-172	151-172#	151-176	151-176#	151-182	151-182#
	151-196	151-196#	151-200	151-200#	151-213	151-213#	151-219	151-219#	152-47	152-47#				
M$TLAB	1-C29#	87-153#	101-17#	101-21#	101-25#	101-29#	101-39#	101-102#	101-106#	101-111#	101-115#	101-120#	101-124#	101-128#
	101-138#	101-186#	102-9#	102-11#	102-78#	102-80#	102-82#	102-84#	105-26#	105-44#	105-47#	105-59#	105-94#	105-112#
	114-35#	116-40#	117-45#	117-56#	121-25#	127-39#	127-43#	128-26#	128-41#	129-20#	129-91#	133-27#	133-35#	133-48#
	136-122#	136-128#	136-134#	137-17#	137-30#	137-33#	137-36#	137-95#	138-30#	138-42#	138-53#	139-14#	141-26#	141-36#
	141-59#	141-61#	141-82#	141-107#	143-46#	143-52#	143-58#	143-65#	143-93#	143-103#	143-105#	143-127#	143-129#	143-140#
	143-146#	143-209#	143-320#	143-331#	143-333#	143-337#	143-351#	143-353#	144-19#	144-21#	144-28#	144-30#	144-42#	145-12#
	146-14#	146-16#	147-12#	148-14#	150-42#	150-114#	150-117#	150-118#	151-16#	151-18#	151-26#	151-33#	151-39#	151-45#
	151-46#	151-49#	151-54#	151-67#	151-68#	151-71#	151-74#	151-76#	151-94#	151-95#	151-96#	151-97#	151-98#	151-99#
	151-100#	151-137#	151-143#	151-147#	151-162#	151-166#	151-172#	151-176#	151-182#	151-196#	151-200#	151-213#	151-219#	152-47#
M$TSTL	1-C21#	87-153#	101-17	101-17#	101-21	101-21#	101-25	101-25#	101-29	101-29#	101-39	101-39#	101-102	101-102#
	101-106	101-106#	101-111	101-111#	101-115	101-115#	101-120	101-120#	101-124	101-124#	101-128	101-128#	101-138	101-138#
	101-186	101-186#	102-9	102-9#	102-9#	102-11	102-11#	102-78	102-78#	102-80	102-80#	102-82	102-82#	102-84
	102-84#	105-26	105-26#	105-26#	105-44	105-44#	105-44#	105-47	105-47#	105-59	105-59#	105-94	105-94#	105-94#
	105-112	105-112#	105-112#	114-35	114-35#	116-40	116-40#	117-45	117-45#	117-56	117-56#	121-25	121-25#	121-25#
	127-39	127-39#	127-43	127-43#	128-26	128-26#	128-41	128-41#	129-20	129-20#	129-91	129-91#	129-91#	133-27
	133-27#	133-35	133-35#	133-35#	133-48	133-48#	133-48#	136-122	136-122#	136-122#	136-128	136-128#	136-128#	136-134
	136-134#	136-134#	137-17	137-17#	137-30	137-30#	137-33	137-33#	137-36	137-36#	137-36#	137-95	137-95#	137-95#
	138-30	138-30#	138-42	138-42#	138-42#	138-53	138-53#	138-53#	139-14	139-14#	141-26	141-26#	141-36	141-36#
	141-59	141-59#	141-61	141-61#	141-82	141-82#	141-107	141-107#	143-46	143-46#	143-52	143-52#	143-58	143-58#
	143-65	143-65#	143-93	143-93#	143-103	143-103#	143-103#	143-105	143-105#	143-127	143-127#	143-129	143-129#	143-140
	143-140#	143-146	143-146#	143-209	143-209#	143-320	143-320#	143-331	143-331#	143-333	143-333#	143-337	143-337#	143-351
	143-351#	143-353	143-353#	144-19	144-19#	144-19#	144-21	144-21#	144-28	144-28#	144-28#	144-30	144-30#	144-42
	144-42#	145-12	145-12#	146-14	146-14#	146-16	146-16#	147-12	147-12#	148-14	148-14#	150-42	150-42#	150-114
	150-114#	150-117	150-117#	150-118	150-118#	151-16	151-16#	151-18	151-18#	151-26	151-26#	151-33	151-33#	151-39
	151-39#	151-45	151-45#	151-46	151-46#	151-49	151-49#	151-54	151-54#	151-67	151-67#	151-68	151-68#	151-71
	151-71#	151-74	151-74#	151-76	151-76#	151-94	151-94#	151-95	151-95#	151-96	151-96#	151-97	151-97#	151-98

Cross reference table (CREF V05.01)