RA60/80/81 UDA50/KDA50-Q BSC SUB

UDA50A KDA50Q CZUDHA0

AH-T929A-MC FICHE

FICHE 01 OF 02

FEB 1985

digital

Made In USA

RA60/80/81 UDA50/KDA50-Q BSC SUB
UDA50A KDA50Q CZUDHA0

AH-T929A-MC FICHE
FICHE 02 OF 02

FEB 1985
digital
Made In USA

IDENTIFICATION

PRODUCT CODE: AC-T928A-MC

PRODUCT NAME: CZUDHAO UDA50A/KDA50Q BASIC SUBSYSTEM DIAGNOSTIC

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 UDA50A or KDA50-Q Disk Controller and the associated disk drives. There are three tests within this diagnostic:

Test 1 - Bus Addressing Test (BAT). Runs the UDA50A or KDA50-Q ROM resident diagnostics, then further tests the bus address interface and controller memory.

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

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

This program is designed to handle all future disk drives that are attached to the UDA50A 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.

The following PDP-11 diagnostic programs are also provided for the UDA50A or KDA50-Q disk subsystem:

CZUDIA0 - UDA50A/KDA50-Q Disk Drive Exerciser.

CZUDKA0 - UDA50A/KDA50-Q Disk Drive 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 UDA50A 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).

VECTOR (O) 154 ?

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

BR LEVEL (D) 5 ?

Answer with the interrupt priority used by the controller. Levels 4 to 7 are accepted. If the controller is a UDA50A, this level must match the level "hard wired" in the controller by the priority plug. If the controller is a KDA50-Q, the answer to this question is ignored and level 4 is used because all Q-bus devices interrupt at level 4.

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).

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 IN TEST 2 (L) N ?

Test 2 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, test 2 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.

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 UDA50A 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 UDA50A with four units.

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

UNIT 1
CSR ADDRESS (O) 172150 ?  <CR>
VECTOR (O) 154 ?  <CR>
BR LEVEL (D) 5 ?  <CR>
DRIVE # (D) 0 ?  <CR>

UNIT 2
CSR ADDRESS (O) 172150 ?  <CR>
VECTOR (O) 154 ?  <CR>
BR LEVEL (D) 5 ?  <CR>
DRIVE # (D) 0 ?  1<CR>

UNIT 3
CSR ADDRESS (O) 172150 ?  <CR>
VECTOR (O) 154 ?  <CR>
BR LEVEL (D) 5 ?  <CR>
DRIVE # (D) 1 ?  2<CR>

UNIT 4
CSR ADDRESS (O) 172150 ?  <CR>
VECTOR (O) 154 ?  <CR>
BR LEVEL (D) 5 ?  <CR>
DRIVE # (D) 2 ?  3<CR>
```

As you can see from the above example, the hardware parameters do not very 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 1
```

```
CSR ADDRESS (O) 172150 ?  <CR>
VECTOR (O) 154 ?  <CR>
BR LEVEL (D) 5 ?  <CR>
DRIVE # (D) 0 ?  0-3<CR>
```

Example 2:

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

UNIT 1
CSR ADDRESS (O) 172150 ?  <CR>
VECTOR (O) 154 ?  <CR>
BR LEVEL (D) 5 ?  <CR>
DRIVE # (D) 0 ?  0,1,2,3<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 CSR address is 172150, the vector is 154, and the BR level is 5 for each entry since they 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 ZUDHA0"
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 UDA50A or KDA50-Q:

```
DR>STA/FLA:PNT/PAS:1

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

UNIT 0
CSR ADDRESS OF CONTROLLER (O)  172150 ?
VECTOR (O)  154 ?
BR LEVEL (D)  5 ?
DRIVE # (D) O? 0,1

CHANGE SW (L)  ? Y

ENTER MANUAL INTERVENTION MODE IN TEST 2 (L) N ?

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

TST: 002
TST: 003

DR>
```

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 UDA50A 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 ZUD^"0" command to the XXDP+ monitor contains two parts, the host level code and three programs which are loaded into the UDA50A or KDA50-Q for execution. These three programs are called "diagnostic machine" or DM programs. The "diagnostic machine" is the facility in the controller which executes a PDP-11 like program. The large majority of the testing is done by these three "diagnostic machine" programs. 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 UDA50A 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 UDA50A or KDA50-Q being tested. It may be omitted if the error is not specific to one controller.

BUS ADDRESSING DM PC: dmpc CONTROLLER AT csr RUNTIME hh:mm:ss

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

DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hh:mm:ss

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

DISK FUNCTIONAL DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hh:mm:ss

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

Sample error message:

```
CZUDH 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.

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, UDA50A 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 00001 - CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL -

```
CZUDH SYS FTL ERR  00001 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
CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL
```

When the hardware questions were answered, two units were selected with the same CSR address but with a different vector or BR level. A single UDA50A or KDA50-Q can have only one vector orBR level. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

3.2.1.2 00002 - TWO UNITS SELECT SAME DRIVE -

```
CZUDH 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.

H2

3.2.1.3 00003 MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER -

```
CZUDH 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 UDA50A or KDA50-Q at one time. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

3.2.1.4 00004 NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED -

```
CZUDH SYS FTL ERR  00004 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  RUNTIME hh:mm:ss
NOT ENOUGH 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.

3.2.1.5 00006 - TABLE CONSISTANCY ERROR -

```
CZUDH SYS FTL ERR  00006 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  RUNTIME hh:mm:ss
TABLE CONSISTANCY 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.

3.2.1.6 00008 TWO CONTROLLERS USE THE SAME VECTOR -

```
CZUDH SYS FTL ERR  00008 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 CONTROLLERS USE THE SAME VECTOR
```

The hardware questions for two units specified different controller CSR addresses but identical vector addresses. The program is aborted and returns to the supervisor prompt so that you can change the hardware questions.

3.2.1.7 00014 - CONTROLLER IS NOT SUPPORTED BY DIAGNOSTIC PROGRAM -

```
CZUDH 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 UDA50A (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 UDA50A 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.

3.2.1.8 00021 - FATAL ERROR REPORTED BY CONTROLLER -

```
CZUDH DVC FTL ERR  00021 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  105154
REPLACE CONTROLLER SDI MODULE
```

The controller resident diagnostic detected a failure. The error is displayed in the SA register. Here are the possible error values and their meaning:

```
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
```

J2

SEQ 0022

122240 - DMA test error
122300 - STEP 3 INIT error
142300 - STEP 4 INIT error

Replace either the processor module or the SDI interface module as specified in the error message.

K2

SEQ 0023

3.2.1.9 00022 - STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION -

```
CZUDH DVC FTL ERR  00022 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION
STEP BIT EXPECTED 004000
SA REGISTER CONTAINS  000000
REPLACE CONTROLLER PROCESSOR MODULE
```

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

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

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

3.2.1.10 00023 - CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY -

```
CZUDH DVC FTL ERR  00023 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY DURING INITIALIZATION
6 WORDS WERE TO BE CLEARED STARTING AT ADDRESS 040644
THE FOLLOWING WORDS NOT CLEARED:
   ADDRESS    CONTENTS
   040644     000010
   040650     000010
   040652     000010
REPLACE CONTROLLER PROCESSOR MODULE
```

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

3.2.1.11 00024 - SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 -

```
CZUDH DVC FTL ERR  00024 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION
PURGE/POLE DIAGNOSTICS WERE REQUESTED
SA REGISTER CONTENTS  004400
```

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

3.2.1.12 00025 - INCORRECT DATA RETURNED IN SA REGISTER -

```
CZUDH DVC FTL ERR  00025 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
INCORRECT DATA RETURNED IN SA REGISTER DURING INITIALIZATION
SA EXPECTED  004400
SA REGISTER CONTAINS  004000
REPLACE CONTROLLER PROCESSOR MODULE
```

For each step of initialization, specific data is expected to be displayed in the SA register. If the contents of the SA register does not match the expected data, then error message 00025 is displayed. Replace controller processor module.

3.2.1.13 00026 - DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST -

```
CZUDH DVC FTL ERR  00026 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST
DATA SENT TO SA REGISTER 000001
RECEIVED FROM SA REGISTER 000000
REPLACE CONTROLLER PROCESSOR MODULE
```

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

M2

SEQ 0025

3.2.1.14 00027 - SA REGISTER DID NOT CHANGE AFTER WRITING TO IT -

```
CZUDH DVC FTL ERR  00027 ON UNIT 00 TST tst SUB sub PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
SA REGISTER DID NOT CHANGE AFTER WRITING TO IT
IN PORT LOOP DIAGNOSTIC
SA REGISTER CONTAINS  004400
REPLACE CONTROLLER PROCESSOR MODULE
```

The controller can be put into a mode where the SA register acts as a wrap port. While the controller is in this mode, any data being sent to the SA register will be displayed in the SA register within a small period of time. If, after the host program sends data to it while it was in diagnostic wrap mode, the controller does not change the contents of the SA register, error message 00027 is displayed. Replace controller processor module.

3.2.1.15 00028 - CONTROLLER DID NOT INTERRUPT THE HOST -

```
CZUDH DVC FTL ERR  00028 ON UNIT 00 TST 001 SUB 005 PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
CONTROLLER DID NOT INTERRUPT THE HOST
REPLACE CONTROLLER PROCESSOR MODULE
```

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

3.2.1.16 00029 - CONTROLLER INTERRUPTED AT A DIFFERENT BR LEVEL THAN EXPECTED -

```
CZUDH DVC FTL ERR  00029 ON UNIT 00 TST 001 SUB 005 PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
CONTROLLER INTERRUPTED AT DIFFERENT BR LEVEL THAN EXPECTED
INTERRUPT WAS AT BR LEVEL 5
```

The controller interrupted at a different BR level than expected. If the controller under test is a UDA50A, be sure the priority plug on the controller and the BR level specified during the hardware questions match. If these are the same, this error indicates a problem with the controller processor module. If the controller is a KDA50-Q, the BR level will always be 4 and this error will indictate a problem with the controller processor module.

3.2.1.17 00030 - FATAL ERROR REPORTED BY CONTROLLER -

```
CZUDH 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
```

3.2.1.18 00031 - NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES

```
CZUDH 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.

3.2.1.19 00032 - UNKNOWN REQUEST RECEIVED FROM DM PROGRAM -

```
CZUDH 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.

3.2.1.20 00033 - RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA -

```
CZUDH 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).

3.2.1.21 00035 - DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE -

```
CZUDH 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.

3.2.1.22 00036 - NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS

```
CZUDH 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.

3.2.1.23 00038 - MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS -

```
CZUDH DVC FTL ERR  00038 ON UNIT 00 TST tst SUB sub 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.

3.2.1.24 00039 - DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST -

```
CZUDH DVC FTL ERR  00039 ON UNIT 00 TST 001 SUB 008 PC: hostpc
HOST PROGRAM  CONTROLLER AT 172150  RUNTIME hh:mm:ss
DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST
CONTROLLER MEMORY ADDRESS xxxxxx
DATA WRITTEN xxxxxx DATA READ xxxxxx
```

Subtest 8 of the Bus Addressing Test performs a controller memory test by writing known data patterns to the controller memory and then reading the data back. If the data read is not the same as the data written this error will occur. This usually indicates a bad memory location in the controller or a controller memory addressing problem.

3.2.2 TEST 1 ERROR MESSAGES (01000 TO 01999) -

3.2.2.1 1000 - DIAGNOSTIC INTERNAL ERROR DETECTED -

```
CZUDH DVC FTL ERR  01000 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
PARAMETERS PASSED TO DIAGNOSTIC FROM HOST ARE INCORRECT

EITHER AT LEAST ONE WORD IS NOT WRITEABLE OR READABLE OR
THE HOST WRITEABLE OR READABLE AREA DOES NOT BEGIN OR END
ON A WORD BOUNDARY
WRITEABLE AREA:
 START                    write_start (HEX)
 END                      write_end   (HEX)
READABLE AREA:
 START                    read_start  (HEX)
 END                      read_end    (HEX)

write_start:
     Starting address of the writeable area in hex.

write_end:
     Ending address of the writeable area in hex.

read_start:
     Starting address of the readable area in hex.

read_end:
     Ending address of the readable area in hex.

Before starting the test, BAT checks the parameters passed to it  from
the host to make sure that all the parameters are valid.  If a problem
is found with a parameter (odd address, no readable or writeable area)
the above error is printed.

This usually indicates a host processor  or  host  memory  with  major
problems.  Run the processor and memory diagnostics before continuing.

NOTE:  BAT requires AT LEAST one word of writeable  and  one  word  of
readable  memory.  To  specify  one  word,  the  starting  and ending
addresses would be the same.
```

3.2.2.2 1001 - NON-EXISTENT MEMORY ERROR DURING ONE-WORD READ -

```
CZUDH DVC FTL ERR  01001 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
NON-EXISTENT MEMORY ERROR DURING ONE-WORD READ FROM HOST MEMORY
                                  OCTAL              HEX
```

ADDRESS octal_address hex_address

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the location of the non-existent memory.

During the reading of the address that contain their own addresses (the cooperating host portion of the test), BAT got a NXM error. NXMs are expected in the readable area, but not in the writeable area. When a NXM is detected during the reading of the addresses, the above error is printed.

This usually indicates the controller has a high addressing line that is stuck at one. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

3.2.2.3 1002 - PARITY ERROR DURING ONE-WORD READ -

```
CZUDH DVC FTL ERR  01002 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
PARITY ERROR DURING ONE-WORD READ FROM HOST MEMORY
                                OCTAL              HEX
ADDRESS                     octal_address      hex_address
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the location of the parity error.

If a parity error is detected during the reading of a address containing its own address (the cooperating host portion of the test) OR the unique addressing test, the above error error is printed.

This can be caused by bad a bad transmitter on the host memory, a bad receiver on the controller, stuck-at-one-or-zero data lines in the controller or a bad backplane. Run the processor and memory diagnostics before continuing.

3.2.2.4 1003 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD READ -

```
CZUDH DVC FTL ERR  01003 ON UNIT unit TST 001 SUB 00C PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD READ
FROM HOST MEMORY
ERROR CODE RETURNED FROM CONTROLLER: code  (DECIMAL)
                                OCTAL              HEX
ADDRESS                     octal_address      hex_address
```

code:
The error code returned in DM R1 that BAT does not recognize.

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the location which caused the unknown error

After every one-word read, BAT checks the contents of DM R1. The only errors that are possible are NXM and parity. If BAT sees any other error code in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

3.2.2.5 1004 - NON-EXISTENT MEMORY ERROR DURING ONE-WORD WRITE -

```
CZUDH DVC FTL ERR  01004 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
NON-EXISTENT MEMORY ERROR DURING ONE-WORD WRITE TO HOST MEMORY
                                OCTAL              HEX
ADDRESS                   octal_address      hex_address
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the location of the non-existent memory.

If, when writing during the unique addressing test, a NXM is detected within the writeable area, the above error is printed.

This usually indicates the controller has a high addressing line that is stuck at one. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

3.2.2.6 1005 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD WRITE -

```
CZUDH DVC FTL ERR  01005 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING ONE-WORD WRITE
TO HOST MEMORY
ERROR CODE RETURNED FROM CONTROLLER: code  (DECIMAL)
                                OCTAL              HEX
ADDRESS                   octal_address      hex_address
```

code:
The error code returned in DM R1 that BAT does not recognize.

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the location which caused the unknown error

After every one-word write, BAT checks the contents of DM R1. The only error possible is a NXM. If BAT sees a error code other than a NXM in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

3.2.2.7 1006 - HOST MEMORY ADDRESSING ERROR. TWO ADDRESSES ACCESS THE SAME LOCATION -

```
CZUDH DVC FTL ERR  01006 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
HOST MEMORY ADDRESSING ERROR.  TWO ADDRESSES ACCESS THE SAME LOCATION
                               OCTAL              HEX
KNOWN GOOD ADDRESS         octal_good         hex_good
TEST ADDRESS               octal_test         hex_test
ADDRESS BIT IN ERROR       octal_xor          hex_xor
```

octal_good, hex_good:
The 32 bit octal and hex address, respectively, of the location which is 'base' of the test.

octal_test, hex_test:
The 32 bit octal and hex address, respectively, of the address that was under test.

octal_xor, hex_xor:
The 32 bit octal and hex values, respectively, showing the addressing bit that is stuck-at-one or stuck-at-zero.

If, during the unique addressing test, the same location is found to respond to two different addresses, the above error is printed.

This indicates that the addressing bit shown on the 'ADDRESS BIT IN ERROR' line is either stuck-at-zero or stuck-at-one.

3.2.2.8 1007 - HOST MEMORY ADDRESSING ERROR. LOCATION READ SHOULD CONTAIN -

```
CZUDH DVC FTL ERR  01007 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
HOST MEMORY ADDRESSING ERROR.  LOCATION READ SHOULD CONTAIN
THE LOW 16 BITS OF ITS ADDRESS
                                OCTAL              HEX
ADDRESS                    octal_address      hex_address
DATA READ                  octal_data         hex_data
DATA EXPECTED              octal_expected     hex_expected
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the location which has been read.

octal_data, hex_data:
The 16 bit octal and hex contents, respectively, of the address above.

octal_expected, hex_expected:
The 16 bit octal and hex values, respectively, which was expected to be found in the address above.

If, while reading the addresses that contain their own addresses, BAT detects that an address DOES NOT contain the low 16 bits of it's own address, the above error is printed.

This usually indicates an addressing bit that is stuck-at-one or stuck-at-zero. Look for error 1006 later on in the error reports. This should give a better indication of the problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

3.2.2.9 1008 - NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER WRITE -

```
CZUDH DVC FTL ERR  01008 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER WRITE TO HOST MEMORY
                                OCTAL              HEX
BUFFER STARTING ADDRESS    octal_address      hex_address
BUFFER SIZE IN BYTES       octal_size         hex_size
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the start of the large buffer write.

octal_size, hex_size:
The 16 bit octal and hex size, respectively, of the transfer.

If BAT detects a NXM during a large buffer write, the above error is

printed (the entire writeable area should writeable and addressable)

This usually indicates the controller has a high addressing line that is stuck at one. Another possibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

3.2.2.10 1009 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER WRITE -

```
CZUDH DVC FTL ERR  01009 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER WRITE
ERROR CODE RETURNED FROM CONTROLLER: code  (DECIMAL)
                                OCTAL              HEX
BUFFER STARTING ADDRESS  octal_address      hex_address
BUFFER SIZE IN BYTES     octal_size         hex_size
```

code:
The error code returned in DM R1 that BAT does not recognize.

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the start of the large buffer write.

octal_size, hex_size:
The 16 bit octal and hex size, respectively, of the transfer.

After every large buffer write, BAT checks the contents of DM R1. The only error possible is a NXM. If BAT sees a error code other than a NXM (or a success) in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another posibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

3.2.2.11 1010 - NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER READ -

```
CZUDH DVC FTL ERR  01010 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
NON-EXISTENT MEMORY ERROR DURING LARGE-BUFFER READ FROM HOST MEMORY
                                OCTAL              HEX
BUFFER STARTING ADDRESS  octal_address      hex_address
BUFFER SIZE IN BYTES     octal_size         hex_size
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the start of the large buffer read.

octal_size, hex_size:
The 16 bit octal and hex size, respectively, of the transfer.

If BAT detects a NXM during a large buffer read, the above error is printed (the entire writeable area should readable)

This usually indicates the controller has a high addressing line that is stuck at one. Another posibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

3.2.2.12 1011 - PARITY ERROR DURING LARGE-BUFFER READ -

```
CZUDH DVC FTL ERR  01011 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
PARITY ERROR DURING LARGE-BUFFER READ FROM HOST MEMORY
                                OCTAL              HEX
BUFFER STARTING ADDRESS  octal_address      hex_address
BUFFER SIZE IN BYTES     octal_size         hex_size
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the start of the large buffer read.

octal_size, hex_size:
The 16 bit octal and hex size, respectively, of the transfer.

While doing a large buffer read, BAT detected a parity error in the data read from the host.

This can be caused by bad a bad transmitter on the host memory, a bad receiver on the controller, stuck-at-one-or-zero data lines in the controller or a bad backplane. Another posibility is that the controller has some kind of stress or heat related problem. Run the processor and memory diagnostics before continuing.

3.2.2.13 1012 - UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER READ -

```
CZUDH DVC FTL ERR  01012 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DIAGNOSTIC INTERNAL ERROR DETECTED
UNKNOWN ERROR CODE RETURNED FROM CONTROLLER DURING LARGE-BUFFER READ
FROM HOST MEMORY
```

```
ERROR CODE RETURNED FROM CONTROLLER: code  (DECIMAL)
                                OCTAL           HEX
BUFFER STARTING ADDRESS  octal_address      hex_address
BUFFER SIZE IN BYTES     octal_size         hex_size
```

code:
The error code returned in DM R1 that BAT does not recognize.

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the start of the large buffer read.

octal_size, hex_size:
The 16 bit octal and hex size, respectively, of the transfer.

After every large buffer read, BAT checks the contents of DM R1. The only errors possible are a NXM and parity error. If BAT sees a error code other than a NXM, parity error (or a success) in DM R1, the above error is printed.

This is usually caused by running old diagnostics on a new and improved controller. Another posibility is that the controller has some kind of stress or heat related problem. A less likely possibility is that the controller has major problems. Except for a very pathological problem, this is unlikely, since with major problems the diagnostic wouldn't run properly.

3.2.2.14 1013 - DATA COMPARE FAILED AFTER LARGE-BUFFER WRITE THEN READ -

```
CZUDH DVC FTL ERR  01013 ON UNIT unit TST 001 SUB 000 PC: hostpc
BUS ADDRESSING  DM PC:dmpc  CONTROLLER AT caddr  DRIVE plug  RUNTIME hh:mm:ss
DATA COMPARE FAILED AFTER LARGE-BUFFER WRITE THEN READ FROM HOST MEMORY
CURRENT DATA PATTERN: pattern
                                OCTAL           HEX
BUFFER STARTING ADDRESS  octal_address      hex_address
BUFFER SIZE IN BYTES     octal_size         hex_size
ADDRESS OF ERROR         octal_error        hex_error
DATA READ                octal_data         hex_data
DATA EXPECTED            octal_expected     hex_expected
```

octal_address, hex_address:
The 32 bit octal and hex address, respectively, of the start of the large buffer read.

octal_size, hex_size:
The 16 bit octal and hex size, respectively, of the transfer.

octal_error, hex_error:
The 32 bit octal and hex address, respectively, of where the data comparison failed.

octal_data, hex_data:
The 16 bit octal and hex contents, respectively, of the address above.

octal_expected, hex_expected:
The 16 bit octal and hex values, respectively, which was expected to be found in the address above.

After a large buffer read, BAT checks the data read against a known pattern. If any difference is found between the pattern read and the known pattern, the above error is printed.

This usually indicates the controller has a data line that is stuck at one or zero. Another possibility is that the controller has some kind of stress or heat related memory problem. A less likely possibility is that the host processor or host memory have major problems. Run the processor and memory diagnostics before continuing.

3.2.3 TEST 2 ERROR MESSAGES (02000 TO 02999) -

3.2.3.1 02000 - HOST SPECIFIED UNIT THAT CAN'T BE FOUND -

```
CZUDH HRD ERR  02000 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
HOST SPECIFIED UNIT #0 THAT CAN'T BE FOUND.
TEST2 RESTARTING
```

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

3.2.3.2 02001 - CANNOT RECEIVE VALID DRIVE STATE FROM DRIVE -

```
CZUDH HRD ERR  02001 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
CANNOT RECEIVE VALID DRIVE STATE FROM DRIVE AFTER DRIVE WAS INITED
CHECK IF DRIVE IS POWERED ON.
```

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

3.2.3.3 02002 - DRIVE STATE RECEIVED HAS BAD PARITY -

```
CZUDH HRD ERR  02002 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE STATE RECEIVED HAS BAD PARITY AFTER DRIVE WAS INITED
```

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

3.2.3.4 02003 - DRIVE IS NOT ASSERTING RECEIVER READY -

```
CZUDH HRD ERR  02003 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE IS NOT ASSERTING RECEIVER READY IN DRIVE STATE AFTER DRIVE WAS INITED
```

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

3.2.3.5 02004 - TIMEOUT ON SEND OF ECHO COMMAND TO DRIVE -

```
CZUDH HRD ERR  02004 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF ECHO COMMAND TO DRIVE
  ECHO DATA  FF
```

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

3.2.3.6 02005 - ERROR DURING RECEIVE OF ECHO RESPONSE FROM DRIVE -

```
CZUDH HRD ERR  02005 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF ECHO RESPONSE FROM DRIVE
  ECHO DATA  FF
```

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

3.2.3.7 02006 - ECHO COMMAND RESPONDED WITH DIFFERENT DATA -

```
CZUDH HRD ERR  02006 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ECHO COMMAND RESPONDED WITH DIFFERENT DATA
  ECHO DATA SENT      00FE
  ECHO DATA RECEIVED  00FF
```

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

3.2.3.8 02007 - ERROR BIT SET IN GET STATUS RESPONSE AFTER DRIVE CLEAR COMMAND -

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

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

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

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

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

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

3.2.3.9 02008 - TIMEOUT ON SEND OF ONLINE COMMAND TO DRIVE -

CZUDH HRD ERR 02008 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF ONLINE COMMAND TO DRIVE

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

3.2.3.10 02009 - ERROR DURING RECEIVE OF ONLINE RESPONSE FROM DRIVE -

CZUDH HRD ERR 02009 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF ONLINE RESPONSE FROM DRIVE
explanation

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

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

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

FRAMING ERROR OCCURED ON SDI LEVEL 0 READ DURING RECEIVE XFC
This is caused by one of the following conditions: 1) Illegal frame code -- the frame is not a message start, continue, or end frame. 2) Illegal sequence of frames -- such as a message start frame without ever receiving a message end frame. This can be caused by the drive sending a response before the 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 OCCURED ON SDI LEVEL 0 READ DURING RECEIVE XFC
The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad controller receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable.

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

CODE FROM RECEIVE XFC WAS UNINTELLIGIBLE FROM SUBSYSTEM 0000

The responce from the drive was not anything that was expected.
Possible controller microcode change without test 2 update.

3.2.3.11 02010 - ONLINE COMMAND WAS UNSUCCESSFUL -

```
CZUDH HRD ERR  02010 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ONLINE COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R 10 L): 1312  1110  0908  0706  0504  0302  0100
```

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

3.2.3.12 02011 - ONLINE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE

```
CZUDH HRD ERR  02011 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ONLINE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
EXPECTED RESPONSE  7E
ACTUAL RESPONSE    00
```

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

3.2.3.13 02012 - TIME-OUT ON SENT OF GET UNIT CHARACTERISTICS COMMAND

```
CZUDH HRD ERR  02012 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF GET UNIT CHARACTERISTICS COMMAND TO DRIVE
```

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

3.2.3.14 02013 - ERROR DURING RECEIVE OF GET UNIT CHARACTERISTICS COMMAND -

```
CZUDH HRD ERR  02013 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF GET UNIT CHARACTERISTICS COMMAND FROM DRIVE
explanation
```

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

3.2.3.15 02014 - GET UNIT CHARACTERISTICS COMMAND WAS UNSUCCESSFUL -

```
CZUDH HRD ERR  02014 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
GET UNIT CHARACTERISTICS COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

3.2.3.16 02015 - GET UNIT CHARACTERISTICS COMMAND DID NOT RETURN EXPECTED RESPONSE -

```
CZUDH HRD ERR  02015 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
GET UNIT CHARACTERISTICS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
  EXPECTED RESPONSE  78
  ACTUAL RESPONSE    00
```

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

3.2.3.17 02016 - HOST PROGRAM GAVE DM CODE IMPROPER DATA -

```
CZUDH HRD ERR  02016 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
HOST PROGRAM GAVE DM CODE IMPROPER DATA
  EXPECTED VALUE SHOULD BE BETWEEN 0 AND 3
  ACTUAL VALUE WAS xx
```

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

3.2.3.18 02017 - TIME-OUT ON SEND OF DIAGNOSE COMMAND TO DRIVE -

```
CZUDH HRD ERR  02017 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF DIAGNOSE COMMAND TO DRIVE
```

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

3.2.3.19 02018 - ERROR DURING RECEIVE OF DIAGNOSE RESPONSE FROM DRIVE -

```
CZUDH HRD ERR  02018 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF DIAGNOSE RESPONSE FROM DRIVE
explanation
```

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

3.2.3.20 02019 - DIAGNOSE COMMAND WAS UNSUCCESSFUL -

```
CZUDH HRD ERR  02019 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DIAGNOSE COMMAND WAS UNSUCCESSFUL
REAL TIME STATE state
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

3.2.3.21 02020 - DIAGNOSE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

```
CZUDH HRD ERR  02020 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DIAGNOSE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
  EXPECTED RESPONSE  FC
  ACTUAL RESPONSE    00
```

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

3.2.3.22 02021 - DRIVE DIAGNOSTIC REPORTS A HARD ERROR -

```
CZUDH HRD ERR  02021 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE DIAGNOSTIC REPORTS A HARD ERROR
  TEST NUMBER  0000
  DRIVE TYPE   00
  ERROR NUMBER 0000
  data
```

The drive diagnostic found an error and is reporting the error back to the host. All values are in hex. TEST NUMBER shows what test was run. DRIVE TYPE shows what type of drive was being tested. ERROR NUMBER shows the result of the test. The drive may pass back data to the host. This data will be presented in a 32 bit hex format following the error message. More data may follow the 32 bit hex values. This data is printed in ascii format. For definitions of what these values mean, refer to the drive functional spec.

3.2.3.23 02024 - TIME-OUT ON SEND OF MEMORY READ COMMAND -

CZUDH HRD ERR 02024 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF MEMORY READ COMMAND TO DRIVE

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

3.2.3.24 02025 - ERROR DURING RECEIVE OF EMEORY READ RESPONSE FROM DRIVE -

CZUDH HRD ERR 02025 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF MEMORY READ RESPONSE FROM DRIVE
explanation

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

3.2.3.25 02026 - MEMORY READ COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02026 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
MEMORY READ COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

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

3.2.3.26 02027 - MEMORY READ COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 02027 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
MEMORY READ COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
EXPECTED RESPONSE 72
ACTUAL RESPONSE 00

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

3.2.3.27 02028 - TIME-OUT ON SEND OF MEMORY WRITE COMMAND TO DRIVE -

CZUDH HRD ERR 02028 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF MEMORY WRITE COMMAND TO DRIVE

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

3.2.3.28 02029 - ERROR DURING RECEIVE OF MEMORY WRITE RESPONSE FROM DRIVE -

CZUDH HRD ERR 02029 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF MEMORY WRITE RESPONSE FROM DRIVE
explanation

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

3.2.3.29 02030 - MEMORY WRITE COMMAND WAS UNSUCCESSFUL -

CZUDH HRD ERR 02030 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
MEMORY WRITE COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

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

3.2.3.30 02031 - MEMORY WRITE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

```
CZUDH HRD ERR  02031 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
MEMORY WRITE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
  EXPECTED RESPONSE  7E
  ACTUAL RESPONSE    00
```

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

3.2.3.31 02032 - TIME-OUT ON SEND OF RUN COMMAND TO DRIVE -

```
CZUDH HRD ERR  02032 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF RUN COMMAND TO DRIVE
```

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

3.2.3.32 02033 - ERROR DURING RECEIVE OF RUN RESPONSE FROM DRIVE -

```
CZUDH HRD ERR  02033 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF RUN RESPONSE FROM DRIVE
explanation
```

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

3.2.3.33 02034 - RUN COMMAND WAS UNSUCCESSFUL -

```
CZUDH HRD ERR  02034 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
RUN COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

3.2.3.34 02035 - RUN COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

```
CZUDH HRD ERR  02035 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
RUN COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
  EXPECTED RESPONSE  7E
  ACTUAL RESPONSE    00
```

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

3.2.3.35 02036 - TIME-OUT ON SEND OF RECALIBRATE COMMAND TO DRIVE -

```
CZUDH HRD ERR  02036 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF RECALIBRATE COMMAND TO DRIVE
```

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

3.2.3.36 02037 - ERROR DURING RECEIVE OF RECALIBRATE RESPONSE FROM DRIVE -

```
CZUDH HRD ERR  02037 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF RECALIBRATE RESPONSE FROM DRIVE
explanation
```

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

3.2.3.37 02038 - RECALIBRATE COMMAND WAS UNSUCCESSFUL -

```
CZUDH HRD ERR  02038 uN UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
RECALIBRATE COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

3.2.3.38 02039 - RECALIBRATE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

```
CZUDH HRD ERR  02039 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
RECALIBRATE COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
  EXPECTED RESPONSE  7E
  ACTUAL RESPONSE    00
```

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

3.2.3.39 02040 - TIME-OUT ON SEND OF GET STATUS COMMAND TO DRIVE -

```
CZUDH HRD ERR  02040 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF GET STATUS COMMAND TO DRIVE
```

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

3.2.3.40 02041 - ERROR DURING RECEIVE OF GET STATUS RESPONSE FROM DRIVE -

```
CZUDH HRD ERR  02041 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF GET STATUS RESPONSE FROM DRIVE
explanation
```

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

3.2.3.41 02042 - GET STATUS COMMAND WAS UNSUCCESSFUL -

```
CZUDH HRD ERR  02042 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
GET STATUS COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

3.2.3.42 02043 - GET STATUS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

CZUDH HRD ERR 02043 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
GET STATUS COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
EXPECTED RESPONSE F6
ACTUAL RESPONSE 00

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

3.2.3.43 02044 - TIME-OUT ON SEND OF DRIVE CLEAR COMMAND TO DRIVE -

CZUDH HRD ERR 02044 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
TIME-OUT ON SEND OF DRIVE CLEAR COMMAND TO DRIVE

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

3.2.3.44 02045 - ERROR DURING RECEIVE OF DRIVE CLEAR RESPONSE FROM DRIVE -

CZUDH HRD ERR 02045 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
ERROR DURING RECEIVE OF DRIVE CLEAR RESPONSE FROM DRIVE
explanation

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

3.2.3.45 02046 - DRIVE CLEAR COMMAND WAS UNSUCCESSFUL -

```
CZUDM HRD ERR  02046 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE CLEAR COMMAND WAS UNSUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 13'2  1110  0908  0706  0504  0302  0100
```

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

3.2.3.46 02047 - DRIVE CLEAR COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

```
CZUDM HRD ERR  02047 ON UNIT 00 TST 002 SUB 000 PC: hostpc
DISK RESIDENT  DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE CLEAR COMMAND DID NOT RETURN EXPECTED RESPONSE CODE
  EXPECTED RESPONSE  7E
  ACTUAL RESPONSE    00
```

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

3.2.4 TEST 3 INFORMATIONAL MESSAGES -

3.2.4.1 LOGGABLE INFORMATION AFTER RECAL -

```
UNIT xx CO?ITROLLER AT csr DRIVE plug   RUNTIME hhh:mm:ss
LOGGABLE INFORMATION AFTER RECAL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

After sending a RECALIBRATE command, the ATTENTION bit was set. Test 3 then sent a GET STATUS command and found the LOGGABLE INFORMATION bit was set. This is not an error, it is only some information being sent from the drive. Normal operation continues. Check error 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5 TEST 3 ERROR MESSAGES (03000 TO 03999) -

3.2.5.1 03001 - TIME-OUT ON SEND -

```
CZUDH HRD ERR  03001 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT ON SEND
COMMAND WAS command
REAL TIME STATE state
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

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

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

The bit positions are defined as follows:
- 0001 - Receiver ready (Test 3 able to transmit to drive)
- 0002 - Attention (error occurred or online timeout expired)
- 0040 - Available (drive offline and usuable)
- 1000 - Read/Write ready

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

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

3.2.5.2 03002 - TIME-OUT OF RECEIVE -

```
CZUDH HRD ERR  03002 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
TIME-OUT OF RECEIVE
COMMAND WAS GET COMMON CHARACTERISTICS
REAL TIME STATE state
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

This error is a failure of the drive to respond to an SDI level 2 command (see the SDI specification) befor the drive-supplied command timeout expires. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.3 03003 FIRST WORD RECEIVED WAS NOT A START FRAME -

```
CZUDH HRD ERR  03003 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
FIRST WORD RECEIVED WAS NOT A START FRAME
COMMAND WAS GET COMMON CHARACTERISTICS
REAL TIME STATE state
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

The first word received by the controller from the drive was not a valid message start frame. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.4 03004 - FRAMING ERROR ON LEVEL 0 RESPONSE -

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

Error 3004 is caused by one or more of the following conditions: 1) Illegal frame code -- the frame is not a message start, continue, or end frame. 2) Illegal sequence of frames -- such as a message start frame without ever receiving a message end frame. This can be caused by the drive sending a response before the controller asserts receiver ready, or a random hit on the SDI cable that garbles a frame or a bad drive transmitter or controller receiver. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.5 03005 - CHECKSUM ERROR ON LEVEL 0 RESPONSE -

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

The checksum attached to a message end frame did not match the checksum computed over the level 2 command. This could be caused by a bad drive transmitter, bad controller receiver, incorrectly computed checksum by the drive (unlikely) or a random hit on the SDI cable. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.6 03006 - RESPONSE LONGER THAN EXPECTED -

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

The buffer size set aside for the response was not large enough for the response received. This is caused by the drive sending a response that is incorrect for the request sent to the drive, or the drive sending some garbage with the response. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.7 03007 - CODE FROM RECEIVE WAS UNINTELLIGIBLE FROM SUBSYSTEM = 0000 -

```
CZUDH HRD ERR  03007 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
CODE FROM RECEIVE WAS UNINTELLIGIBLE FROM SUBSYSTEM = 0000
COMMAND WAS GET COMMON CHARACTERISTICS
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

The unknown error code occurs when the controller returns an error code from an operation that test 3 does not recognize. Possible controller microcode change without test 3 update. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.8 03008 - COMMAND DID NOT RETURN EXPECTED RESPONSE CODE -

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

This is caused by receiving an UNSUCCESSFUL response from the drive, or the drive sending some response other than the correct response for the request sent to the drive. See the contents of status for the unexpected response error (or reason). Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.9 03009 - DRIVE NOT ASSERTING RECEIVER READY IN DRIVE STATE -

```
CZUDH HRD ERR  03009 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE NOT ASSERTING RECEIVER READY IN DRIVE STATE
REAL TIME STATE 0002
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

Test 3 inits the drive and checks the drive's real time state. If RECEIVER READY was not asserted after a period of time this error message is printed. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.10 03011 - NO VALID STATE FROM DRIVE -

```
CZUDH HRD ERR  03011 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
NO VALID STATE FROM DRIVE
NO DRIVE CLOCKS
CHECK THAT DRIVE IS POWERED ON.
```

If test 3 attempts to get the drive state, and finds that there are no drive clocks on the port, the above message is occurrs. This error usually means that the SDI cable is not connected, the drive is not powered on or the drive's port button that connects it to this controller is not depressed.

3.2.5.11 03012 - NO VALID STATE FROM DRIVE -

CZUDH HRD ERR 03012 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
NO VALID STATE FROM DRIVE
HARD PARITY OR PULSE ERROR FOR 1/2 A SECOND

If test 3 attempts to get the drive state, and gets pulse or parity errors for a full 1/2 second, the above message is printed. This error usually indicates a poor connection or grounding of the SDI cables, a bad drive transmitter, a bad controller receiver or a broken SDI cable.

3.2.5.12 03014 - SUBUNIT CHARACTERISTICS SAY THERE ARE ZERO READ ONLY GROUPS -

CZUDH HRD ERR 03014 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
SUBUNIT CHARACTERISICS SAY THERE ARE ZERO READ ONLY GROUPS
IN THE DIAGNOSTIC AREA

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

3.2.5.13 03015 - LESS THAN 1 READ/WRITE GROUP IN DIAGNOSTIC AREA -

CZUDH HRD ERR 03015 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
SUBUNIT CHARACTERISTICS SAY THERE ARE LESS THAN 1 READ/WRITE
GROUPS IN THE DIAGNOSTIC AREA

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

3.2.5.14 03016 - NEITHER R/W READY NOR ATTENTION SET AFTER RECALIBRATE COMMAND -

CZUDH HRD ERR 03016 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
NEITHER R/W READY NOR ATTENTION SET AFTER RECALIBRATE COMMAND
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

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

transmitter or receiver or by a hit on the SDI cable. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.15 03017 - LESS THAN 1 DIAGNOSTIC CYLINDER -

```
CZUDH HRD ERR  03017 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
SUBUNIT CHARACERISTICS SAY LESS THAN 1 DIAGNOSTIC CYLINDER
```

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

3.2.5.16 03018 - READ/WRITE READY DROPPED BEFORE FORMAT OPERATION -

```
CZUDH HRD ERR  03018 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
READ/WRITE READY DROPPED BEFORE FORMAT OPERATION
CYLINDER aaa.  GROUP bb.  TRACK cc.
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

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

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

3.2.5.17 03019 - FORMAT OPERATION REPORTED TIME-OUT FAILURE -

```
CZUDH HRD ERR  03019 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
FORMAT OPERATION REPORTED TIME-OUT FAILURE
CYLINDER aaa.  GROUP bb.  TRACK cc.
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

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

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

3.2.5.18 03020 - AFTER RECAL, ERROR BITS WERE SET -

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

After sending a RECALIBRATE command, the ATTENTION bit was set. Test 3 then sent a GET STATUS command and found the error bits were set. For further information, check the state and the status. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.19 03022 - READ/WRITE READY DROPPED BEFORE WRITE OPERATION -

```
CZUDH HRD ERR  03022 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
READ/WRITE READY DROPPED BEFORE WRITE OPERATION
CYLINDER aaa.  GROUP bb.  TRACK cc.
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

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

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

3.2.5.20 03023 - COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK

```
CZUDH HRD ERR  03023 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK.  ON LAST BLOCK:
WRITE OPERATION REPORTED FAILURE -- ERROR CODE aaa OCTAL.
```

```
UBN bbb.  CYLINDER ccc.  GROUP dd.  TRACK ee.
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

Where:

aaa is the error code in octal.
It may have one of the following values:

- 2 = drive failure
- 3 = requested LBN is a secondary revector.

<<< NOTE >>> We are working with DBN's

- 4 = header compare failure (desired header not found)
- 133 = suspected positioner error
- 213 = read/write ready failure
- 253 = drive data or state clock timeout (indicates cable/transmitter/receiver broken)
- 313 = receiver ready timeout
- 413 = drive state receive error during write

bbb is the DBN in decimal.
ccc is the cylinder value in decimal.
dd is the group value in decimal.
ee is the track value in decimal.
Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.21 03024 - READ/WRITE READY DROPPED BEFORE READ OPERATION -

```
CZUDH HRD ERR  03024 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
READ/WRITE READY DROPPED BEFORE READ OPERATION
CYLINDER aaa.  GROUP bb.  TRACK cc.
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

Where:

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

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

3.2.5.22 03025 - READ OPERATION REPORTED FAILURE -

```
CZUDH HRD ERR  03025 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK.  ON LAST BLOCK:
READ OPERATION REPORTED FAILURE -- ERROR CODE aaa OCTAL.
CYLINDER ccc.  GROUP dd.  TRACK ee.
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

After each track in the diagnostic space is formatted, at least one block must be able to have data written to it and read from it and the data must be correct. No block from track (ee) was able to pass. The error code (aaa) gives the reason for the read operation failure.

Where:

aaa is the error code in octal.
It may have one of the following values:

- 2 = drive failure
- 3 = requested LBN is a secondary revector.

<<< NOTE >>> We are working with DBN's

- 4 = header compare failure (desired header not found)
- 52 = SERDES overrun error
- 150 = data sync timeout on read
- 153 = suspected positioner error
- 213 = read/write ready failure
- 253 = drive data or state clock timeout (indicates cable/transmitter/receiver broken)
- 313 = receiver ready timeout
- 413 = drive state receive error during write

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

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

3.2.5.23 03026 - COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK

```
CZUDH HRD ERR  03026 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
COULD NOT WRITE AND READ ANY BLOCK ON THIS TRACK. ON LAST BLOCK:
DATA COMPARE FAILURE ON WORD aa.
EXPECTED DATA  bbbb
ACTUAL DATA    cccc
```

CYLINDER ddd. GROUP ee. TRACK ff.

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

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

3.2.5.24 03027 - SEEK COMPLETE TIME-OUT -

CZUDH HRD ERR 03027 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
SEEK COMPLETE TIME-OUT -- READ/WRITE READY DID NOT SET
SEEK WAS TO CYLINDER aaa. GROUP bb.
REAL TIME STATE 0003
STATUS (R TO L): 1312 1110 0908 0706 0504 0302 0100

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

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

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

3.2.5.25 03028 - NO BLOCK ON THIS TRACK CAN BE READ -

CZUDH HRD ERR 03028 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc CONTROLLER AT csr DRIVE plug RUNTIME hhh:mm:ss
NO BLOCK ON THIS TRACK CAN BE READ. LAST BLOCK TRIED:
aBN bbbb. CYLINDER ccc. GROUP dd. TRACK ee.

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

Where:
a is 'L' for LBN, 'D' for DBN, or 'X' for XBN.

bbbb is the block number in decimal.
ccc is the cylinder in decimal.
dd is the group number in decimal.
ee is the track number in decimal.

3.2.5.26 03029 - AVAILABLE WAS NOT ASSERTED AFTER DISCONNECT -

```
CZUDH HRD ERR  03029 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
AVAILABLE WAS NOT ASSERTED AFTER DISCONNECT
REAL TIME STATE 0003
STATUS (R T  L): 1312  1110  0908  0706  0504  0302  0100
```

After the DISCONNECT command was sent, the AVAILABLE flag should be asserted after a period of time. It it never was, then error 3029 appears. There maybe a problem with a transmitter or a receiver or the SDI cable at this point. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.27 03030 - INVALID LEVEL 2 COMMAND OPCODE Aaaa WAS SUCCESSFUL -

```
CZUDH HRD ERR  03030 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
INVALID LEVEL 2 COMMAND OPCODE aaaa WAS SUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

Some invalid level 2 commands are sent over the SDI. The drive should find these illegal commands and flag them as such. If the drive doesn't, then error 3030 will appear. "aaaa" is the invalid command in hex. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.28 03031 - COMMAND WITH Type LENGTH = A WAS SUCCESSFUL -

```
CZUDH HRD ERR  03031 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
COMMAND WITH type LENGTH = a WAS SUCCESSFUL
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

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

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

3.2.5.29 03032 - UNIT DID NOT REPORT TRANSMISSION ERROR -

```
CZUDH HRD ERR  03032 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
UNIT DID NOT REPORT TRANSMISSION ERROR
WHEN reason
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

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

Where reason could be one of the following:

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

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

3.2.5.30 03033 - UNIT ACCEPTED AN INVALID GROUP NUMBER -

```
CZUDH HRD ERR  03033 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
UNIT ACCEPTED AN INVALID GROUP NUMBER FROM GROUP SELECT LEVEL 1
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

A level 1 select group command with an illegal group number is sent to the drive. If the drive accepted it, then error 3033 will be displayed. Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.31 03035 - SUCCESSFULLY WROTE ON DBN AREA WHEN DRIVE WAS WRITE PROTECTED -

```
CZUDH DVC FTL ERR  03035 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
SUCCESSFULLY WROTE ON DBN AREA WHEN DRIVE WAS WRITE PROTECTED
REAL TIME STATE 0003
STATUS (R TO L): 1312  1110  0908  0706  0504  0302  0100
```

An attempt was made to write on a write protected drive. It should

have resulted in an error response from the disk drive, but it didn't.
Check 03001 for explanation of 'REAL TIME STATE' and 'STATUS'

3.2.5.32 03036 - DRIVE NOT PROPERLY FORMATTED -

```
CZUDH DVC FTL ERR  03036 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE IS NOT PROPERLY FORMATTED.
CONTROLLER WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION
THIS DRIVE NEEDS TO BE FORMATTED.
```

Test 3 reads a copy of the FCT in the XBN area and determined that the FCT was corrupted. Any normal operating system (which uses the UDA50A or KDA50-Q as a controller) will spin down the drive, so the drive will need to be reformatted.

3.2.5.33 03037 - DRIVE IS FORMATTED IN 576 BYTE MODE -

```
CZUDH DVC FTL ERR  03037 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
DRIVE IS FORMATTED IN 576 BYTE MODE.
TO RUN WITH A controller, THIS DRIVE NEEDS TO BE FORMATTED IN 512 BYTE MODE.
CONTROLLER WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION
THIS DRIVE NEEDS TO BE FORMATTED.
```

Test 3 reads a copy of the FCT from the XBN area and determined that the drive was formatted in 576 byte mode. Any normal operating system (which uses the UDA50A or KDA50-Q as a controller) will spin down the drive, so the drive will need to be reformatted.

3.2.5.34 03038 - NO COPY OF THE FCT COULD BE READ -

```
CZUDH DVC FTL ERR  03038 ON UNIT 00 TST 003 SUB 000 PC: hostpc
DISK FUNCTION DM PC: dmpc  CONTROLLER AT csr  DRIVE plug  RUNTIME hhh:mm:ss
NO COPY OF THE FCT COULD BE READ.
CONTROLLER WILL SPIN DOWN THIS DRIVE IF USED IN NORMAL SYSTEM OPERATION
THIS DRIVE NEEDS TO BE FORMATTED.
```

Test 3 attempted to read every copy of the FCT without success. Any normal operating system (which uses the UDA50A or KDA50-Q as a controller) will spin down the drive, so the drive will need to be reformatted

3.2.6 SPECIAL DEVICE FATAL (05000) -

```
CZUDH DVC FTL ERR 05000 ON UNIT 00 TST 002 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.

3.3 DEC STANDARD 166 EXCERPTS

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

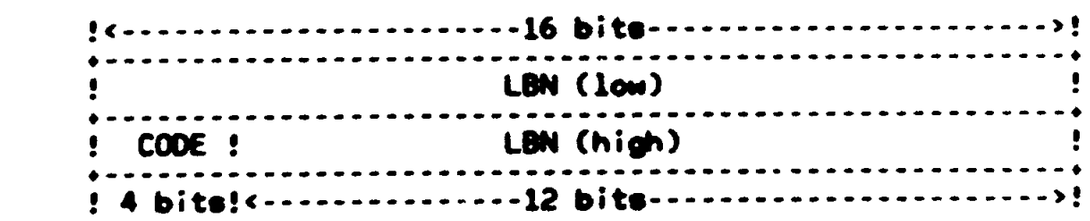
Replacement And Caching Table Format -

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

NOTE

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

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



Where:

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

CODE is one of the following octal values:

- 00 - Unallocated (empty) replacement block.
- 02 - Allocated replacement block - primary RBN.
- 03 - Allocated replacement block - non-primary RBN.
- 04 - Unusable replacement block.
- * 05 - Alternate unusable replacement block
- 10 - Null entry - no corresponding RBN sector.

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

NOTE

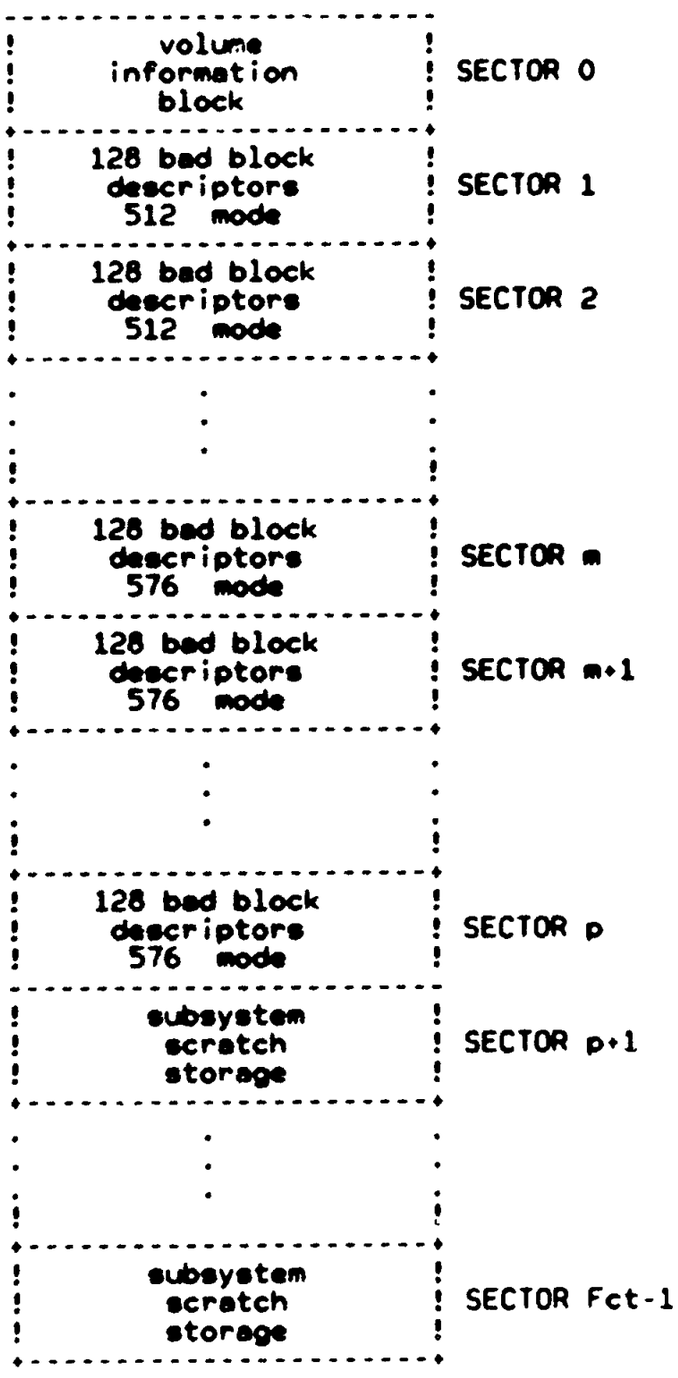
* This code is reserved. Programs should treat this code as if it were code 04.

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

1. Code 02 if the replacement block can be retrieved with little degradation of performance for all blocks.

2. Code 03 if accessing the replacement block has a large impact on performance for all blocks.

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

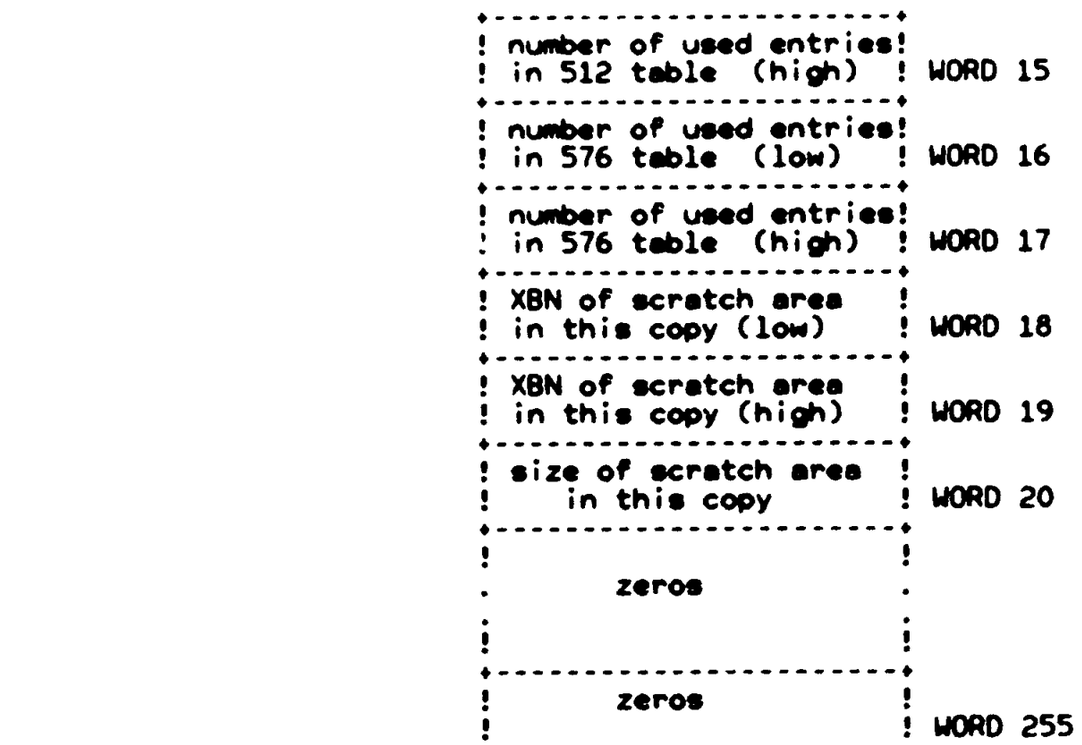


The XBN area itself is always formatted to contain 512 byte sectors. The calculations for m and p are:

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

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

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



Where:

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

4.0 PERFORMANCE AND PROGRESS REPORTS

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

A statistical report will automatically be printed periodically (approximately every fifteen minutes). 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 look like the following example:

TEST 1 IN PROGRESS RUN TIME 2:24:10

5.0 TEST SUMMARIES

The controller Host Resident Diagnostic consists of one PDP-11 diagnostic supervisor program that runs in the PDP-11 processor and three programs that run 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" programs into the controller and starting their execution. The "diagnostic machine" programs control the testing from that point by requesting the PDP-11 processor to supply information, print error messages and update statistics. The "diagnostic machine" programs inform the PDP-11 processor when a test is complete.

5.1 TEST 1 - BUS ADDRESSING TEST

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

1. Check that every address line of the Unibus can be driven to both one and zero states

2. Check that the controller can interrupt the PDP-11 processor at the proper priority level and vector.

3. Exercise the Unibus interface by transferring blocks of data to and from Unibus memory.

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

1. All data bits can be written and read correctly.

2. NPR cycles can be executed correctly.

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

Subtest 1 makes sure that the controller IP and SA registers exist.

Subtest 2 initializes the controller through steps 1 and 2, and runs the first part of the controller's resident diagnostics.

Subtest 3 initializies the controller into diagnostic wrap mode. In this mode any value written into the SA register is echoed in the SA register.

Subtest 4 initializes the controller using the smallest size ring buffer allowed and interrupts disabled. This allows the controller

Resident Diagnostics to perform bus address testing. Any controller Resident Diagnostic errors will be reported.

In subtest 5, the controller is initialized with interrupts enabled. The vector address and priority level will be determined solely from the answers to the hardware questions. If the hardware vectors to the wrong address, it is impossible to determine the result. A descriptive error message of the problem will not occur (the program or processor may hang or an unrelated message may occur). Therefore, the message "TESTING INTERRUPT ABILITY OF CONTROLLER AT ADR csr VEC vec ..." is printed just before the controller is requested to cause an interrupt and the word "COMPLETED" is printed (on the same line) when the interrupt test is completed. If the word "COMPLETED" does not follow the first message, it should be apparent that the interrupt caused the diagnostic or processor to go astray. The priority level of the interrupt request is also verified.

Subtest 6 initializes the controller with a large ring buffer area and interrupts disabled. The large ring buffer areais supplied to allow the controller Resident Diagnostic to do the most bus address testing possible. Any controller Resident Diagnostic errors will be reported.

Subtest 7 downline loads the DM ADDRESSING TEST program into the controller and executes it. The DM addressing test tests the controller's ability to correctly address host memory. In order to test this ability, the DM addressing test consists of three unique parts. Each part is designed to catch a specific set of problems; once the test is complete, the user will have confidence that the controller will be able to correctly address all of host memory.

Part 1 - Read host memory with host cooperation
Before the DM addressing test is started, the host identifies two unique regions of memory: 1 - Readable memory (usually the host identifies all addresses 0 - 2**18 - 2 as readable -- the DM test will tolerate NXMs) 2 - Readable AND writeable memory (usually this is a small area that the host sets aside for the unique use of the DM addressing test). Both areas MUST begin and END on word boundaries (NOTE: the word defined by the ending address IS used).

The host, just prior to starting the =DM addressing test, writes into each word of the WRITEABLE area the address of each location; for example, location 12564 will contain the number 12564, location 12566 will contain 12566, etc. If the writeable area is located in addressing space greater than 2**16 - 2, only the low order 16 bits are stored in the word.

Once the writeable area is filled with the addresses of each location, the DM addressing test is downline-loaded into the controller. The DM test will first ask the host for the bounding addresses of readable and writeable memory. The DM will then read, one word at a time, each word in the writeable area, and compare its contents against the address that the DM attempted to read. If any mismatch is found between an address read and its contents, an error is printed.

Once the entire writeable area is read, Part 1 is complete.

Part 2 - Unique addressing test
The unique addressing test will drive as many addressing lines as possible to both one and zero, checking to make sure that as each one changes, a different and unique location is addressed. This will catch any addressing lines that may be stuck-at-one or stuck-at-zero.

The method used is as follows:

The first word in the writeable area is written with all ones. Now bit 1 of the writeable location's address is toggled. If the resulting address falls within the readable area, the resulting address is read. If a NXM results, or the contents of that address is NOT all ones, the test knows that it has accessed a unique and different location. If the contents of the read location IS all ones, the test then writes all zeroes to the first word of the writeable area. Now it reads the 'toggled' address again. If the contents are NOT zero, the test knows it has accessed a unique location. If the contents of the read location ARE zero, addressing bit 1 is know to be stuck, and an error is reported.

The above process is repeated for bits 2, 3, etc. up to bit 31. However, the 'toggled' address is NOT read from unless it falls within the readable area. For this reason, the readable area should be made as large as possible to facilitate testing.

Part 3 - Large buffer reads and writes
In the third part, the DM addressing test tries to do large buffer writes and reads to and from host memory. This is to stress the controller by causing it to do large transfers at a high speed.

The test determines if the internal buffer memory is either larger or smaller than the host writeable area. If the internal buffer is smaller, it will do several writes in order to fill up the host writeable area. Otherwise, it will scale the writes down to the size of the writeable area.

Three patterns will be written and read to and from host memory. The first pattern will be written until the entire writeable area is filled. Then the writeable area will be read back, and a data compare will be run against the data read. This will be repeated for the next two patterns. Once this test is complete, the entire DM addressing test is finished.

Subtest 8 performs extensive maintenance write and read operations between the host and the controller. Each location in the controller memory is written and read 32 times with a "shifting 1" and "shifting 0" data pattern to check for "stuck at 1" and "stuck at 0" errors. After each location in controller memory is read, it is overwritten with a different pattern to verify the memory is being addressed properly. The data read from the controller is compared to the pattern written to the controller. Any differences found indicate

errors.

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

5.2 TEST 2 - DISK RESIDENT DIAGNOSTIC TEST

The purpose of test 2 is to execute the diagnostics that run in each disk drive. These diagnostic programs are resident in the disk drive. This controller diagnostic program only knows the procedure to execute the disk resident diagnostics and how to determine whether a test passed or failed.

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

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

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

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

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

```
TEST #2 MANUAL INTERVENTION ON UNIT unit CONTROLLER AT csr DRIVE plug
TO WRITE AND READ MEMORY:
  W DATA REGION OFFSET
  R REGION OFFSET
TO RUN A DIAGNOSTIC:
  D REGION
TO EXIT QUESTIONING:
  E
DATA, REGION AND OFFSET ARE HEX VALUES.
 ?
```

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

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

Examples:

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

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

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

5.3 TEST 3 - DISK FUNCTION TEST

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

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

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

1. Issue a DRIVE CLEAR command.

2. Issue RECALIBRATE command.

3. Issue a CHANGE MODE command to enable diagnostic cylinder access, set the drive to 512 byte sector size, and write protect.

4. Issue INITIATE SEEK command to last diagnostic cylinder.

5. Read all factory formatted sector headers. If no headers on a track can be read, report the error, otherwise continue.

6. Starting with cylinder 0, group 0 and incrementing through every cylinder on the disk, seek to a group, read a header on track 0 and then seek to the factory formatted diagnostic cylinder. Read from the diagnostic cylinder to verify disk positioned correctly.

7. Attempt to write on the first diagnostic cylinder while write protected.

8. Issue a CHANGE MODE command to enable formatting operations and disable write protect.

9. Format all writable DBNs in 512 byte format.

10. Write and read several data patterns to each writable DBN. Report an error if all DBNs on one track have an error.

11. Send invalid SDI level 2 and level 1 commands and check the results.

12. Go to the XBN area and read a copy of the FCT. Check to see if the drive has been properly formatted in 512 byte mode.

13. Issue a DISCONNECT command.

@

```
                        .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
```

```
                                        .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  CZUDH,A,0,0,0,PRI07     ; TEST 1-3
                                L$NAME::                        ;DIAGNOSTIC NAME
002000      103                         .ASCII /C/
002001      132                         .ASCII /Z/
002002      125                         .ASCII /U/
002003      104                         .ASCII /D/
002004      110                         .ASCII /H/
002005      000                         .BYTE   0
002006      000                         .BYTE   0
002007      000                         .BYTE   0
002010                          L$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$HPCP::                        ;POINTER TO H.W. QUES.
002016  024662                          .WORD   L$HARD
002020                          L$SPCP::                        ;POINTER TO S.W. QUES.
002020  025012                          .WORD   L$SOFT
002022                          L$HPTP::                        ;PTR. TO DEF. H.W. PTABLE
002022  002134                          .WORD   L$HW
002024                          L$SPTP::                        ;PTR. TO S.W. PTABLE
002024  002146                          .WORD   L$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$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                          .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000340                          .WORD   PRI07
002044                          L$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$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  002342                          .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  021334                          .WORD   L$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  002366                          .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                          EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  002150                          .WORD   L$ERRTBL
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  021500                          .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022676                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022674                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021472                          .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
```

```
      1                                 .SBTTL  DISPATCH TABLE
      2
      3                                 ;++
      4                                 ;       THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
      5                                 ;       IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST. THE
      6                                 ;       ARGUMENT OF 'DISPATCH' INDICATES THE NUMBER OF HARDWARE TESTS
      7                                 ;       IN THE DIAGNOSTIC.
      8                                 ;--
      9
     10 002122                                  DISPATCH 3
        002122  000003                          .WORD   3
        002124                          L$DISPATCH::
        002124  022724                          .WORD   T1
        002126  024524                          .WORD   T2
        002130  024622                          .WORD   T3
     12
```

```
                                .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.
                                ;--

002132                                  BGNHW   DFPTBL
002132  000004                         .WORD   L10000-L$HW/2
002134                          L$HW::
002134                          DFPTBL::
002134  172150                          .WORD   172150                  ; CSR ADDRESS
002136  000154                          .WORD   154                     ; VECTOR ADDRESS
002140  000005                          .WORD   5.                      ; BR LEVEL
002142  000000                          .WORD   0.                      ; LOGICAL DRIVE NUMBER

002144                                  ENDHW
002144                          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
002144  000001                        .WORD   L10001-L$SW/2
002146                          L$SW::
002146                          SFPTBL::
002146  040400                        .WORD   ^B0100000100000000      ;SINGLE BIT QUESTIONS

                                        ENDSW
002150                          L10001:
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

                                        EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
000000 100000                   BIT15== 100000
000000 040000                   BIT14== 40000
000000 020000                   BIT13== 20000
000000 010000                   BIT12== 10000
000000 004000                   BIT11== 4000
000000 002000                   BIT10== 2000
000000 001000                   BIT09== 1000
000000 000400                   BIT08== 400
000000 000200                   BIT07== 200
000000 000100                   BIT06== 100
000000 000040                   BIT05== 40
000000 000020                   BIT04== 20
000000 000010                   BIT03== 10
000000 000004                   BIT02== 4
000000 000002                   BIT01== 2
000000 000001                   BIT00== 1
                                ;
000000 001000                   BIT9==  BIT09
000000 000400                   BIT8==  BIT08
000000 000200                   BIT7==  BIT07
000000 000100                   BIT6==  BIT06
000000 000040                   BIT5==  BIT05
000000 000020                   BIT4==  BIT04
000000 000010                   BIT3==  BIT03
000000 000004                   BIT2==  BIT02
000000 000002                   BIT1==  BIT01
000000 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
                                ;
000000 000340                   PRI07== 340
000000 000300                   PRI06== 300
000000 000240                   PRI05== 240
000000 000200                   PRI04== 200
000000 000140                   PRI03== 140
000000 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                  HOE==  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  = 100000                        ; PURGE POLL TEST ENABLE
                        ;
                        ;       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
```

```
                        .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
0C0004                  HC.RSZ  = 4.                    ; SIZE OF RING IN BYTES
000004                  HC.ESZ  = 4.                    ; SIZE OF ENVELOPE WORDS BEFORE PACKE
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
```

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

                                ;
                                ;       END PACKET FLAGS
                                ;

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

                                ;
                                ;       CONTROLLER FLAGS
                                ;

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

                                ;
                                ;       UNIT FLAGS
                                ;

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

```
                        .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.SHDW  = 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
                        ;
```

COMMAND PACKET OFFSETS

```
                58
                59          000024          P.DMDT  = 20.                   ; DMDT TERMINAL ADDR. (MAINT WRITE ONLY)
                60          000034          P.OVRL  = 28.                   ; BUFFER DESCRIPTOR FOR OPERLAYS
                61
```

```
                        .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.HSTI  = 16.                   ; HOST IDENTIFIER
000024                  P.UNTI  = 20.                   ; UNIT IDENTIFIER
000034                  P.MEDI  = 28.                   ; MEDIA TYPE IDENTIFIER
000040                  P.SHUN  = 32.                   ; SHADOW UNIT
000042                  P.SHST  = 34.                   ; SHADOW STATUS
000044                  P.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.HSTI  = 16.                   ; HOST IDENTIFIER
000024                  P.UNTI  = 20.                   ; UNIT IDENTIFIER
000034                  P.MEDI  = 28.                   ; MEDIA TYPE IDENTIFIER
000040                  P.SHUN  = 32.                   ; SHADOW UNIT
000042                  P.SHST  = 34.                   ; SHADOW STATUS
000044                  P.UNSZ  = 36.                   ; UNIT SIZE
000050                  P.VSER  = 40.                   ; VOLUME SERIAL NUMBER

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

END PACKET OFFSETS

```
                                ;
000014          P.VRSN  = 12.                   ; MSCP VERSION
000016          P.CNTF  = 14.                   ; CONTROLLER FLAGS
000020          P.CTMO  = 16.                   ; CONTROLLER TIMEOUT
000022          P.CSVR  = 18.                   ; CONTROLLER SOFTWARE VERSION
000023          P.CHVR  = 19.                   ; CONTROLLER HARDWARE VERSION
000024          P.CNTI  = 20.                   ; CONTROLLER ID

                                ;
                                ;       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
002150                          ITEM C.UADR     2       <BUS ADDRESS OF IP REGISTER>
002150                          ITEM C.UNIT     2       <UNIT NUMBER TO TEST>
002150                          ITEM C.VEC      2       <VECTOR ADDRESS/BR LEVEL>
002150                          ITEM C.JSR      2       <INTERRUPT SERVICE ROUTINE FOR CONTROLLER>
002150                          ITEM C.JAD      2       <THESE TWO WORDS LOADED WITH [JSR R0    CNTSRV]>
002150                          ITEM C.FLG      2       <FLAGS>
002150                          ITEM C.HCOM     2       <BEGINNING ADRS OF HOST COMM AREA IN MEMORY>
002150                          ITEM C.DR0      2       <POINTER TO DRIVE TABLES>
002150                          ITEM C.DR1      2       <IF ZERO, NO DRIVE TABLE EXISTS>
002150                          ITEM C.DR2      2
002150                          ITEM C.DR3      2
002150                          ITEM C.TO       2       <TIMEOUT COUNTER>
002150                          ITEM C.TOH      2       <  (TWO WORDS)>
002150                          ITEM C.REF      2       <COMMAND REFERENCE NUMBER>

002150                          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

002150                          TABLE                                   ; START A TABLE DEFINITION
002150                          ITEM D.DRV      2       <DRIVE NUMBER>
002150                          ITEM D.UNIT     2       <LOGICAL UNIT NUMBER>

002150                          END D.SIZE      <SIZE OF DRIVE TABLE IN BYTES>

                                ;
                                ;       DM PROGRAM HEADER DEFINITIONS
                                ;

000000                          DMTRLN  = 0                             ; OFFSET TO DOWNLINE LOAD PROGRAM SIZE
000004                          DMOVRL  = 4                             ; OFFSET TO SIZE OF OVERLAY
000040                          DMMAIN  = 40                            ; OFFSET TO FIRST WORD OF MAIN PROGRAM
001000                          DMFRST  = 1000                          ; ADDRESS START OF HEADER IN DM FILE
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

                                        ERRTBL
002150                          L$ERRTBL::
002150  000000                  ERRTYP::        .WORD   0
002152  000000                  ERRNBR::        .WORD   0
002154  000000                  ERRMSG::        .WORD   0
002156  000000                  ERRBLK::        .WORD   0

002160                          FFREE:: .BLKW 1                         ; FIRST FREE WORD IN MEMORY
002162                          FSIZE:: .BLKW 1                         ; SIZE OF FREE MEMORY IN WORDS
002164                          FMEM:   .BLKW 1                         ; COPY OF FFREE AT END OF INIT SECTION
002166                          FMEMS:  .BLKW 1                         ; COPY OF FSIZE AT END OF INIT SECTION
002170                          DTABS:: .BLKW 1                         ; START OF DRIVE TABLE STORAGE
002172                          CTABS:: .BLKW 1                         ; START OF CONTROLLER TABLE STORAGE
002174                          CTRLRS: .BLKW 1                         ; COUNT OF CONTROLLERS IN PTABLES
002176                          TSTTAB: .BLKW 1                         ; POINTER TO 1ST CONTROLLER TABLE
002200                          DMPROG: .BLKW 1                         ; START ADDRESS OF DM PROGRAM

002202                          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

002204  000000                  TNUM:   .WORD 0                         ; NUMBER OF TEST EXECUTING
002206                          URUN:   .BLKW 1                         ; NUMBER OF UNITS TO RUN AT ONE TIME
002210                          URNING: .BLKW 1                         ; NUMBER OF UNITS STILL RUNNING
002212                          UCNT:   .BLKW 1                         ; COUNTER OF UNITS UNDER TEST
002214                          INTRCV: .BLKW 1                         ; INTERRUPT RECEIVED FLAG

002216                          TEMP:   .BLKW 12.                       ; TEMPORY STORAGE FOR GMANI RESPONSES

002246  000000                  IPADRS: .WORD   0                       ; 4 ENTRIES
002250  000000                          .WORD   0                       ;       FOR CONTROLLER
002252  000000                          .WORD   0                       ;               CSR
002254  000000                          .WORD   0                       ;                   ADDRESSES


                                ;
                                ;       KW11 CLOCK CONTROL
                                ;

002256  000000                  KW.CSR: .WORD 0                         ; CSR OF CLOCK
002260                          KW.BRL: .BLKW 1                         ; BR LEVEL
002262                          KW.VEC: .BLKW 1                         ; VECTOR
002264                          KW.HZ:  .BLKW 1                         ; HERTZ (50. OR 60.)
002266                          KW.EL:  .BLKW 2                         ; ELAPSED TIME
002272                          STIME:  .BLKW 2                         ; STATISTICAL REPORT TIMER

002276                          NXMAD:  .BLKW 1                         ; SET TO -1 BY NON-EXISTANT ADDRESS
```

GLOBAL DATA SECTION

```
                                ;
002300                  T2WRR:  .BLKW 1                 ; WRITE/READ REGION
002302                  T2WRO:  .BLKW 1                 ; WRITE/READ OFFSET
002304                  T2DR:   .BLKW 1                 ; DIAGNOSE REGION

002306  012252          PTYPE:  .WORD   PF              ; PRINT TYPE
002310  000             TTYOUT: .BYTE   0               ; TTY OUTPUT BUFFER
002311  000                     .BYTE   0               ; TERMINATOR FOR ASCIZ STRING
                                .EVEN


                        ;
                        ;       DATA TO BE SENT AND RECEIVED BY CONTROLLER INITIALIZATION
                        ;

002312  020500          INITBL: .WORD   RSP.S1          ; 1ST WORD RESPONSE CHECK ROUTINE
002314  000000          SND.S1: .WORD   0               ; 1ST WORD TO SEND TO SA REGISTER
002316  020512                  .WORD   RSP.S2          ; 2ND WORD RESPONSE CHECK ROUTINE
002320  000000          SND.S2: .WORD   0               ; 2ND WORD TO SEND TO SA REGISTER
002322  020532                  .WORD   RSP.S3          ; 3RD WORD RESPONSE CHECK ROUTINE
002324  000000          SND.S3: .WORD   0               ; 3RD WORD TO SEND TO SA REGISTER
                        ;;      .WORD   RSP.S4          ; 4TH WORD RESPONSE CHECK ROUTINE

002326  000000          SSTEP4: .WORD   0               ; LOCATION TO SAVE STEP 4 VALUE
002330  000000          CNTRSD: .WORD   0               ; LOCATION FOR STEP BIT MASK
002332                  WCHNGD: .BLKW   1               ; SAVED LOOPBACK DATA
002334                  BRLEV:  .BLKW   1               ; WORD FOR BR LEVEL STORAGE

        036413          T18MSZ = 36413                  ; CONTROLLER MEMORY SIZE
002336                  T18PAT: .BLKW   1               ; STORAGE FOR DATA PATTERN
002340                  DMPTR:  .BLKW   1               ; STORAGE FOR DM PROGRAM POINTER
```

```
                                .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
                                ;

 002342                                 DEVTYP <LOGICAL DISK DRIVE>
 002342                         L$DVTYP::
 002342  114  117  107                  .ASCIZ  /LOGICAL DISK DRIVE/
                                        .EVEN


                                ;
                                ;       TEST DESCRIPTION
                                ;

 002366                                 DESCRIPT <CZUDHAO UDA50A,KDA50-Q BASIC SUBSY>
 002366                         L$DESC::
 002366  103  132  125                  .ASCIZ  /CZUDHAO UDA50A,KDA50-Q BASIC SUBSY/
                                        .EVEN


                                ;
                                ;       UNFORMATTED MESSAGES
                                ;

 002432  040  040  000          T4OPT7: .ASCIZ \  \


                                ;
                                ;       FORMAT STATEMENTS USED IN PRINT CALLS
                                ;

 002435  045  124  000          FRMTT:  .ASCIZ \%T\
 002440  045  116  000          CRLF:   .ASCIZ \%N\
 002443  042  040  040          RNTIM:  .ASCIZ \"  RUNTIME "D+6":"\
 002466  104  071  042          RNTIM1: .ASCIZ \D9":"\
 002474  104  071  000          RNTIM2: .ASCIZ \D9\
 002477  042  040  040          ERRME1: .ASCIZ \"  *  *  *  ERROR PROCESSING MESSAGE STRING  *  *  *"N\
 002566  116  042  122          MXFERP: .ASCIZ \N"REACHED TRANSFER LIMIT - TESTING STOPPED"N\
 002643  116  042  125          ERRLIM: .ASCIZ \N"UNIT "D6" REACHED ERROR LIMIT - UNIT DROPPED FROM TEST"N\
 002736  116  042  124          INTST0: .ASCIZ \N"TESTING INTERRUPT ABILITY OF CONTROLLER AT ADR "O16" VEC "O9"..."\
 003042  042  103  117          INTST1: .ASCIZ \"COMPLETED"N\
 003057  116  042  115          T2WARN: .ASCIZ \N"MANUAL INTERVENTION NOT ALLOWED, TEST 2 RUNNING UNATTENDED"N\
 003156  116  042  124          T2CMS1: .ASCII \N"TEST 2 MANUAL INTERVENTION ON UNIT "D8" CONTROLLER AT "O16" DRIVE "D9N\
 003266  042  124  117                  .ASCII \"TO WRITE AND READ MEMORY:"N\
 003322  042  040  040                  .ASCII \"  W DATA REGION OFFSET"N\
 003353  042  040  040                  .ASCII \"  R REGION OFFSET"N\
 003377  042  124  117                  .ASCII \"TO RUN A DIAGNOSTIC:"N\
 003426  042  040  040                  .ASCII \"  D REGION"N\
 003443  042  124  117                  .ASCII \"TO EXIT QUESTIONING:"N\
 003472  042  040  040                  .ASCII \"  E"N\
 003500  042  104  101                  .ASCIZ \"DATA, REGION AND OFFSET ARE HEX VALUES."N\
 003553  042  077  040          T2CMS5: .ASCIZ \"? INPUT ERROR"N\
```

GLOBAL TEXT SECTION

```
     66 003574      116      042      125  MESSG:  .ASCIZ \N"UNIT "D6" CONTROLLER AT "016" DRIVE "D9S\
     67 003647      042      116      117  NOCLOCK:.ASCIZ \"NO LINE CLOCK AVAILABLE FOR TIMING EVENTS"N\
     68
     69 003724      042      110      117  BASN0:  .ASCIZ \"HOST PROGRAM"\
     71 003743      042      102      125  BASN1:  .ASCIZ \"BUS ADDRESSING"\
     72 003764      042      104      111  BASN2:  .ASCIZ \"DISK RESIDENT"\
     73 004004      042      104      111  BASN3:  .ASCIZ \"DISK FUNCTION"\
     78 004024      042      040      040  BASL1:  .ASCIZ \"  DM PC:"012\
     79 004042      042      040      040  BASL2:  .ASCIZ \"  CONTROLLER AT "016\
     80 004070      042      040      040  BASL3:  .ASCIZ \"  DRIVE "D9\
     81 004105      000                   BAS:    .BYTE 0                         ; NULL TO PRINT NOTHING
     82
     83 004106      122      066      122  BASLN:  .ASCIZ \R6R6R6R6\               ; FORMAT FOR BASIC LINE OF ERROR MESSAGE
     84
     85 004117                             X1A:
     86 004117                             X2A:
     87 004117                             X3A:
     88 004117      042      111      116  X8A:    .ASCIZ \"INVALID ANSWERS GIVEN TO THE HARDWARE QUESTIONS"N\
     90 004202      122      065      122  X1:     .ASCIZ \R5R6"CONTROLLER HAS MORE THAN ONE VECTOR OR BR LEVEL"N\
     92 004271      122      065      122  X2:     .ASCIZ \R5R6"TWO UNITS SELECT THE SAME DRIVE"N\
     93 004340      122      065      122  X3:     .ASCIZ \R5R6"MORE THAN 4 DRIVES SELECTED ON THIS CONTROLLER"N\
     94 004426      122      064      042  X4:     .ASCII \R4"NOT ENOUGH MEMORY TO TEST THE UNITS SELECTED"N\
     95 004507      042      122      105          .ASCIZ \"RESTART PROGRAM AND TEST FEWER UNITS AT A TIME"N\
     96 004571      122      064      042  X6:     .ASCIZ \R4"TABLE CONSISTANCY ERROR. PLEASE RE-LOAD PROGRAM"N\
     98 004656      122      065      122  X8:     .ASCIZ \R5R6"TWO CONTROLLERS USE THE SAME VECTOR"N\
    100 004731      122      065      042  X14:    .ASCII \R5"CONTROLLER IS NOT SUPPORTED BY THIS DIAGNOSTIC PROGRAM."N\
    101 005025      042      124      110          .ASCII \"THIS PROGRAM REQUIRES A UDA50A (MODEL 6, MICROCODE VERSION"\
    102 005121      042      040      101          .ASCII \" AT LEAST 3)."N\
    103 005141      042      117      122          .ASCII \"OR A KDA50-Q (MODEL 13, MICROCODE VERSION AT LEAST 0)"N\
    104 005231      116      042      103          .ASCIZ \N"CONTROLLER REPORTED MODEL CODE "D4" AND MICROCODE VERSION "D4N\
    106 005332      122      065      042  X22:    .ASCII \R5"STEP BIT DID NOT SET IN SA REGISTER DURING INITIALIZATION"N\
    107 005430      042      123      124          .ASCIZ \"STEP BIT EXPECTED "016NR8R7\
    108 005465      122      065      042  X23A:   .ASCII \R5"CONTROLLER DID NOT CLEAR RING STRUCTURE IN HOST MEMORY "\
    109 005560      042      104      125          .ASCII \"DURING INITIALIZATION"N\
    110 005610      104      071      042          .ASCII \D9" WORDS WERE TO BE CLEARED STARTING AT ADDRESS "016N\
    111 005676      042      124      110          .ASCII \"THE FOLLOWING WORDS WERE NOT CLEARED:"N\
    112 005746      123      066      042          .ASCIZ \S6"ADDRESS"S4"CONTENTS"N\
    113 005777      123      067      117  X23B:   .ASCIZ \S7016S5016N\
    114 006013      122      065      042  X24:    .ASCII \R5"SA REGISTER DID NOT GO TO ZERO AFTER STEP 3 WRITE OF INITIALIZATION"N\
    115 006123      042      120      125          .ASCIZ \"PURGE/POLE DIAGNOSTICS WERE REQUESTED"NR8R7\
    116 006200      122      065      042  X25:    .ASCII \R5"INCORRECT DATA RETURNED IN SA REGISTER DURING INITIALIZATION"N\
    117 006301      042      123      101          .ASCIZ \"SA EXPECTED: "016NR8R7\
    118 006331      122      065      042  X26:    .ASCII \R5"DATA COMPARISON ERROR DURING DIAGNOSTIC PORT LOOP TEST"N\
    119 006424      042      104      101          .ASCII \"DATA SENT TO SA REGISTER: "016N\
    120 006464      042      122      105          .ASCIZ \"RECEIVED FROM SA REGISTER: "016NR7\
    121 006530      122      065      042  X27:    .ASCII \R5"SA REGISTER DID NOT CHANGE AFTER WRITING TO IT"N\
    122 006613      042      111      116          .ASCIZ \"IN PORT LOOP DIAGNOSTIC"NR8R7\
    123 006652      122      065      042  X28:    .ASCIZ \R5"CONTROLLER DID NOT INTERRUPT THE HOST"NR7\
    124 006727      122      065      042  X29:    .ASCII \R5"CONTROLLER INTERRUPTED AT DIFFERENT BR LEVEL THAN EXPECTED"N\
    125 007026      042      111      116          .ASCIZ \"INTERRUPT WAS AT BR LEVEL "03N\
    127 007066      122      065      042  X31:    .ASCII \R5"NO INTERRUPT RECEIVED FROM DM PROGRAM FOR 3 MINUTES"N\
    128 007156      042      101      123          .ASCIZ \"ASSUME PROGRAM IS HUNG"N\
    129 007210      122      065      042  X32:    .ASCIZ \R5"MESSAGE BUFFER RECEIVED FROM DM PROGRAM WITH UNKNOWN REQUEST NUMBER"N\
    130 007321      122      065      042  X35:    .ASCIZ \R5"DM PROGRAM ASKED FOR DATA ON UNKNOWN DRIVE"N\
    131 007401      122      065      042  X36:    .ASCII \R5"NO INTERRUPT RECEIVED FROM CONTROLLER FOR 30 SECONDS"N\
    132 007472      042      127      110          .ASCIZ \"WHILE LOADING DM PROGRAM"N\
    133 007526      122      065      042  X38:    .ASCII \R5"MEMORY ERROR TRYING TO READ CONTROLLER REGISTERS"N\
```

```
    134 007613      042     103     110             .ASCII \"CHECK BUS SELECTION SWITCHES ON CONTROLLER PROCESSOR MODULE"N\
    135 007711      042     117     122             .ASCII \"OR BUS"N\
    136 007722      042     117     122             .ASCIZ \"OR "R7\
    138 007732      122     065     042     X39:    .ASCII \R5"DATA COMPARE ERROR DURING CONTROLLER MEMORY TEST"N\
    139 010017      042     103     117             .ASCII \"CONTROLLER MEMORY ADDRESS: "O16N\
    140 010060      042     104     101             .ASCIZ \"DATA WRITTEN: "O16" DATA READ: "O16N\
    146 010126      122     065     042     XFCE:   .ASCIZ \R5"FATAL ERROR REPORTED BY CONTROLLER"NR8\
    147 010200      042     115     105     XMSG1:  .ASCIZ \"MESSAGE BUFFER CONTAINS:"N\
    148 010234      123     063     117     XMSG2:  .ASCIZ \S3016S1016S1016S1016S1016S1016S1016N\
    149 010301      122     065     042     XPKT1:  .ASCII \R5"RESPONSE PACKET FROM CONTROLLER DOES NOT CONTAIN EXPECTED DATA"N\
    150 010404      042     105     111             .ASCII \"EITHER CONTROLLER RETURNED ERROR STATUS OR PACKET WAS RECEIVED INCORRECTLY"
N\
    151 010521      123     063     042             .ASCIZ \S3"COMMAND PACKET SENT"S6"RESPONSE PACKET RECEIVED"N\
    152 010606      123     066     117     XPKT2:  .ASCIZ \S6016S1016S14016S1016N\
    153 010635      042     123     101     XSA:    .ASCIZ \"SA REGISTER CONTAINS: "O16N\
    154 010672      042     122     105     XFRU:   .ASCIZ \"REPLACE CONTROLLER PROCESSOR MODULE"N\
    155 010741      042     122     105     XFRUA:  .ASCIZ \"REPLACE CONTROLLER SDI 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.
                                ;--

011002                          BGNMSG ERR001
011002                          ERR001::
011002                                  PNTB    X1,#X1A
011016                          ENDMSG
011016                          L10002:
011016  104423                          TRAP    C$MSG

011020                          BGNMSG ERR002
011020                          ERR002::
011020                                  PNTB    X2,#X2A
011034                          ENDMSG
011034                          L10003:
011034  104423                          TRAP    C$MSG

011036                          BGNMSG ERR003
011036                          ERR003::
011036                                  PNTB    X3,#X3A
011052                          ENDMSG
011052                          L10004:
011052  104423                          TRAP    C$MSG

011054                          BGNMSG ERR004
011054                          ERR004::
011054                                  PNTB    X4
011064                          ENDMSG
011064                          L10005:
011064  104423                          TRAP    C$MSG

011066                          BGNMSG ERR006
011066                          ERR006::
011066                                  PNTB    X6
011076                          ENDMSG
011076                          L10006:
011076  104423                          TRAP    C$MSG

011100                          BGNMSG ERR008
011100                          ERR008::
011100                                  PNTB    X8,#X8A
011114                          ENDMSG
011114                          L10007:
011114  104423                          TRAP    C$MSG

011116                          BGNMSG ERR014
011116                          ERR014::
011116                                  PNTB    X14,R3,R1
011132                          ENDMSG
011132                          L10010:
011132  104423                          TRAP    C$MSG
```

```
 42 011134                              BGNMSG ERR021
    011134                              ERR021::
 43 011134                                      PNTB    XFCE,R2
 44 011146  010201                              MOV     R2,R1
 45 011150  000301                              SWAB    R1
 46 011152  042701  177775                      BIC     #^C<2>,R1
 47 011156  001405                              BEQ     ERR21A
 48 011160                                      PNTB    XFRUA
 49 011170  000404                              BR      ERR21E
 50 011172                              ERR21A:
 51 011172                                      PNTB    XFRU
 52 011202                              ERR21E:
 53 011202                              ENDMSG
    011202                              L10011:
    011202  104423                             TRAP    C$MSG
 54
 55 011204                              BGNMSG ERR022
    011204                              ERR022::
 56 011204  042737  100000  002330              BIC     #SA.ERR,CNTRSD
 57 011212                                      PNTB    X22,CNTRSD,R2
 58 011230                              ENDMSG
    011230                              L10012:
    011230  104423                             TRAP    C$MSG
 59
 60 011232                              BGNMSG ERR023
    011232                              ERR023::
 61 011232                                      PNTB    X23A,R1,FFREE
 62 011250  005742                              TST     -(R2)
 63 011252  005712                      ERR23A: TST     (R2)
 64 011254  001406                              BEQ     ERR23B
 65 011256                                      PNTB    X23B,R2,(R2)
 66                                     ;;      DEC     R4
 67                                     ;;      BEQ     ERR23C
 68 011272  005722                      ERR23B: TST     (R2)+
 69 011274  005303                              DEC     R3
 70 011276  001365                              BNE     ERR23A
 71                                     ;;ERR23C:
 72 011300                                      PNTB    XFRU
 73 011310                              ENDMSG
    011310                              L10013:
    011310  104423                             TRAP    C$MSG
 74
 75 011312                              BGNMSG ERR024
    011312                              ERR024::
 76 011312                                      PNTB    X24,R2
 77 011324                              ENDMSG
    011324                              L10014:
    011324  104423                             TRAP    C$MSG
 78
 79 011326                              BGNMSG ERR025
    011326                              ERR025::
 80 011326                                      PNTB    X25,R1,R2
 81 011342                              ENDMSG
    011342                              L10015:
    011342  104423                             TRAP    C$MSG
 82
 83 011344                              BGNMSG ERR026
```

```
     011344                             ERR026::
  84 011344                                     PNTB    X26,2(R4),R2
  85 011362                             ENDMSG
     011362                             L10016:
     011362  104423                             TRAP    C$MSG
  86
  87 011364                             BGNMSG ERR027
     011364                             ERR027::
  88 011364                                     PNTB    X27,2(R4)
  89 011400                             ENDMSG
     011400                             L10017:
     011400  104423                             TRAP    C$MSG
  90
  91 011402                             BGNMSG ERR028
     011402                             ERR028::
  92 011402                                     PNTB    X28
  93 011412                             ENDMSG
     011412                             L10020:
     011412  104423                             TRAP    C$MSG
  94
  95 011414                             BGNMSG ERR029
     011414                             ERR029::
  96 011414                                     PNTB    X29,R1
  97 011426                             ENDMSG
     011426                             L10021:
     011426  104423                             TRAP    C$MSG
  99
 100 011430                             BGNMSG ERR030
     011430                             ERR030::
 101 011430                                     PNTB    XFCE,R1
 102 011442                             ENDMSG
     011442                             L10022:
     011442  104423                             TRAP    C$MSG
 103
 104 011444                             BGNMSG ERR031
     011444                             ERR031::
 105 011444                                     PNTB    X31
 106 011454                             ENDMSG
     011454                             L10023:
     011454  104423                             TRAP    C$MSG
 107
 108 011456                             BGNMSG ERR032
     011456                             ERR032::
 109 011456                                     PNTB    X32
 110 011466  004737  011660                     CALL    MSGPKT
 111 011472                             ENDMSG
     011472                             L10024:
     011472  104423                             TRAP    C$MSG
 112
 113 011474                             BGNMSG ERR033
     011474                             ERR033::
 114 011474  004737  011566                     CALL    PNTPKT
 115 011500                             ENDMSG
     011500                             L10025:
     011500  104423                             TRAP    C$MSG
 116
 117 011502                             BGNMSG ERR035
```

```
     011502                             ERR035::
118  011502                                     PNTB    X35
119  011512  004737  011660                     CALL    MSGPKT
120  011516                             ENDMSG
     011516                             L10026:
     011516  104423                             TRAP    C$MSG
121
122  011520                             BGNMSG ERR036
     011520                             ERR036::
123  011520                                     PNTB    X36
124  011530                             ENDMSG
     011530                             L10027:
     011530  104423                             TRAP    C$MSG
125
126  011532                             BGNMSG ERR038
     011532                             ERR038::
127  011532                                     PNTB    X38
128  011542                             ENDMSG
     011542                             L10030:
     011542  104423                             TRAP    C$MSG
129
131  011544                             BGNMSG ERR039
     011544                             ERR039::
132  011544                                     PNTB    X39,R1,T18PAT,(R2)
133  011564                             ENDMSG
     011564                             L10031:
     011564  104423                             TRAP    C$MSG
140
141  011566                             PNTPKT: PNTB    XPKT1
142  011576  010401                             MOV     R4,R1
143  011600  062701  000020                     ADD     #HC.CPK,R1
144  011604  010402                             MOV     R4,R2
145  011606  062702  000020                     ADD     #HC.MPK,R2
146  011612  012703  000014                     MOV     #12.,R3
147
148  011616                             PNTPKL: PNTB    XPKT2,2(R1),(R1),2(R2),(R2)
149  011642  062701  000004                     ADD     #4,R1
150  011646  062702  000004                     ADD     #4,R2
151  011652  005303                             DEC     R3
152  011654  001360                             BNE     PNTPKL
153  011656  000207                             RETURN
154
155  011660                             MSGPKT: PNTB    XMSG1
156  011670  016504  000014                     MOV     C.HCOM(R5),R4
157  011674  062704  000206                     ADD     #HC.BF2,R4
158  011700  012703  000005                     MOV     #5,R3
159  011704                             MSGPKL: PNTB    XMSG2,(R4),2(R4),4(R4),6(R4),8.(R4),10.(R4),12.(R4)
160  011746  062704  000016                     ADD     #14.,R4
161  011752  005303                             DEC     R3
162  011754  001353                             BNE     MSGPKL
163  011756  000207                             RETURN
164
165  011760                             BGNMSG ERR.TN                          ; ERROR REPORT ROUTINE
     011760                             ERR.TN::
166  011760  013702  002204                     MOV     TNUM,R2                ; GET TEST NUMBER
167  011764  006302                             ASL     R2                     ; DOUBLE
168  011766  012703  004070                     MOV     #BASL3,R3              ; GET ADDRESS OF DRIVE PRINT LINE
```

```
    169 011772  005764  000004                  TST     4(R4)                   ; CHECK IF DRIVE NUMBER GIVEN
    170 011776  100002                          BPL     1$                      ;       BRANCH IF SO
    171 012000  012703  004105                  MOV     #BAS,R3
    172 012004                          1$:     PNTB    BASLN,TNAMES-2(R2),#BASL1,(R4),#BASL2,(R5),R3,4(R4)
    173 012042                                  ASSUME C.UADR EQ 0
    174 012042  004737  021046                  CALL    RNTIME                  ; GET RUNTIME PARAMETERS
    175 012046                                  PRINT   #CR                     ; ADVANCE TO NEW LINE
    176 012056  062704  000006                  ADD     #6,R4                   ; SET R4 TO POINT TO MESSAGE POINTER
    177 012062  012402                          MOV     (R4)+,R2                ; GET MESSAGE POINTER
    178 012064  006302                          ASL     R2                      ; DOUBLE TO MAKE BYTE OFFSET
    179 012066  063702  002200                  ADD     DMPROG,R2               ; ADD TO START OF MESSAGE STRINGS
    180 012072  067702  170102                  ADD     $DMPROG,R2              ; ADD SIZE OF MAIN PROGRAM
    181 012076  105712                          TSTB    (R2)                    ; CHECK FIRST BYTE
    182 012100  001001                          BNE     NCON                    ; IF ZERO
    183 012102  005202                          INC     R2                      ;       INCREMENT TO NEXT BYTE
    184 012104  012737  012322  002306  NCON:   MOV     #PX,PTYPE               ; CHANGE TO EXTENDED OUTPUT
    185 012112  004737  012476                  CALL    OSTRNG                  ; OUTPUT ACCORDING TO STRING
    186 012116                          ENDMSG
        012116                          L10032:
        012116  104423                          TRAP    C$MSG
    187
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

012120                          FMERR:  ERRSF 4.,ERR004
012120  104454                          TRAP    C$ERSF
012122  000004                          .WORD   4
012124  000000                          .WORD   0
012126  011054                          .WORD   ERR004
                                                                ; DO CLEAN-UP TRAP
012130                                  DOCLN
012130  104444                          TRAP    C$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.
                                ;--

012132                          ALOCM:  PUSH    <FFREE>                 ; SAVE FFREE AT ENTRY
012136  160137  002162                  SUB     R1,FSIZE                ; REDUCE SIZE OF FREE MEMORY
012142  002766                          BLT     FMERR                   ; REPORT ERROR IF NOT ENOUGH MEMORY
012144  060101                          ADD     R1,R1                   ; CHANGE WORDS TO BYTES
012146  060137  002160                  ADD     R1,FFREE                ; CALCULATE NEW START OF FREE MEMORY
012152                                  POP     <R1>                    ; GET START OF ALLOCATED MEMORY
012154  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
                                ;--

012156  012701  000146          HCOMM:  MOV     #<HC.SIZ>/2,R1          ; GET SIZE OF AREA TO ALLOCATE
012162  004737  012132                  CALL    ALOCM                   ; ALLOCATE THE MEMORY
```

GLOBAL SUBROUTINES SECTION

```
     012166  010104                         MOV     R1,R4                   ; GET ADDRESS OF HOST COMM AREA
     012170  010465  000014                 MOV     R4,C.HCOM(R5)           ; PLACE BEGINNING ADRS OF HOST COMM
                                                                            ;       AREA IN THE CONTROLLER TABLE
     012174  062701  000020                 ADD     #HC.MPK,R1              ; COMPUTE START OF MESSAGE PACKET
     012200  010164  000004                 MOV     R1,HC.MSG(R4)           ; PLACE IN RING
     012204  062701  000000                 ADD     #<HC.CPK-HC.MPK>,R1     ; COMPUTE START OF COMMAND PACKET
     012210  010164  000010                 MOV     R1,HC.CMD(R4)           ; PLACE IN RING
     012214  000207                         RETURN

                                    ;++
                                    ;       PPINTC - PRINT A CHARACTER
                                    ;
                                    ;       CALL WITH MACRO PRINT
                                    ;--

     012216  110037  002310         PRINTC: MOVB    R0,TTYOUT               ; SAVE CHARACTER FOR TTY OUTPUT
     012222                                 PUSH    <R1>
     012224  012701  002435                 MOV     #FRMTT,R1               ; GET FORMATTED ASCIZ STRING STATEMENT
     012230  120027  000015                 CMPB    R0,#CR                  ; IF NOT A CARRIAGE RETURN, THEN
     012234  001002                         BNE     1$                      ; PRINT SOME OTHER CHARACTER, ELSE
     012236  012701  002440                 MOV     #CRLF,R1                ; GET FORMATTED ASCIZ STRING STATEMENT
                                                                            ;       GO PRINT CR-LF.
     012242  004777  170040         1$:     CALL    @PTYPE                  ; PRINT THE ASCIZ STRING.
     012246                                 POP     <R1>
     012250  000207                         RETURN
     012252                         PF:     PRINTF  R1,#TTYOUT
     012252  012746  002310                 MOV     #TTYOUT,-(SP)
     012256  010146                         MOV     R1,-(SP)
     012260  012746  000002                 MOV     #2,-(SP)
     012264  010600                         MOV     SP,R0
     012266  104417                         TRAP    C$PNTF
     012270  062706  000006                 ADD     #6,SP
     012274  000207                         RETURN
     012276                         PB:     PRINTB  R1,#TTYOUT
     012276  012746  002310                 MOV     #TTYOUT,-(SP)
     012302  010146                         MOV     R1,-(SP)
     012304  012746  000002                 MOV     #2,-(SP)
     012310  010600                         MOV     SP,R0
     012312  104414                         TRAP    C$PNTB
     012314  062706  000006                 ADD     #6,SP
     012320  000207                         RETURN
     012322                         PX:     PRINTX  R1,#TTYOUT
     012322  012746  002310                 MOV     #TTYOUT,-(SP)
     012326  010146                         MOV     R1,-(SP)
     012330  012746  000002                 MOV     #2,-(SP)
     012334  010600                         MOV     SP,R0
     012336  104415                         TRAP    C$PNTX
     012340  062706  000006                 ADD     #6,SP
     012344  000207                         RETURN
     012346                         PS:     PRINTS  R1,#TTYOUT
     012346  012746  002310                 MOV     #TTYOUT,-(SP)
     012352  010146                         MOV     R1,-(SP)
     012354  012746  000002                 MOV     #2,-(SP)
     012360  010600                         MOV     SP,R0
     012362  104416                         TRAP    C$PNTS
     012364  062706  000006                 ADD     #6,SP
     012370  000207                         RETURN
```

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

012372  012737  012252  002306  LPNTF:  MOV     #PF,PTYPE
012400  000413                          BR      LPNT

012402  012737  012276  002306  LPNTB:  MOV     #PB,PTYPE
012410  000407                          BR      LPNT

012412  012737  012322  002306  LPNTX:  MOV     #PX,PTYPE
012420  000403                          BR      LPNT

012422  012737  012346  002306  LPNTS:  MOV     #PS,PTYPE

012430                          LPNT:   PUSH    <R2,R3,R4,R5>
012440  012102                          MOV     (R1)+,R2                ; GET ADDRESS OF ASCIZ STRING
012442  010604                          MOV     SP,R4                   ; COMPUTE ADDRESS OF 1ST ARGUMENT AND
012444  062704  000012                  ADD     #12,R4                  ; SAVE IT IN R4.
012450                                  PUSH    <R1>                    ; SAVE R1 ON THE STACK
012452  004737  012476                  CALL    OSTRNG                  ; PRINT THE FORMATTED MESSAGE
012456                                  POP     <R0,R5,R4,R3,R2,R1>     ; RESTORE REGISTERS FROM STACK
012472  062006                          ADD     (R0)+,SP                ; ADJUST STACK POINTER OVER ARGUMENTS
012474  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
                                ;--

012476  112201                  OSTRNG: MOVB    (R2)+,R1                ; SEE IF TERMINATOR IN ASCIZ STRING.
012500  001421                          BEQ     OSTRE                   ; EXIT
012502  012700  013006                  MOV     #ERRC,R0                ; GET POINTER TO CHARACTER TABLE
012506  120110                  NCONS:  CMPB    R1,(R0)                 ; COMPARE CHARACTER WITH TABLE ENTRY
012510  001407                          BEQ     NCONF                   ; BRANCH IF MATCH FOUND
012512  105720                          TSTB    (R0)+                   ; INCREMENT POINTER
012514  001374                          BNE     NCONS                   ; CONTINUE SEARCH IF NOT END OF TABLE
012516                                  PNTF    ERRME1                  ; REPORT BAD CONTROL CHARACTER
012526  000406                          BR      OSTRE                   ;
012530  162700  013006          NCONF:  SUB     #ERRC,R0                ; GET INCREMENT INTO TABLE
012534  006300                          ASL     R0                      ; DOUBLE TO WORD COUNT
012536  004770  013020                  CALL    @ERRD(R0)               ; DISPATCH TO PRINT ROUTINE
012542  000755                          BR      OSTRNG                  ; GET NEXT
012544  000207                  OSTRE:  RETURN

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

012546  112200                  CON.QU: MOVB    (R2)+,R0                ; GET CHARACTER
012550  120027  000042                  CMPB    R0,#'"                  ; CHECK IF ENDING QUOTE
012554  001403                          BEQ     CON.QX                  ; IF SO, GO GET NEXT CONTROL CHARACTER
012556                                  PRINT   R0
012562  000771                          BR      CON.QU                  ;       CONTINUE PRINTING
012564  000207                  CON.QX: RETURN

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

012566  004737  015474          CON.A:  CALL    GETCNT                  ; GET COUNT OF CHARACTERS
012572                          CON.A1: PRINT   (R4)+                   ;
012600  005301                          DEC     R1                      ; COUNT THE CHARACTERS
012602  001373                          BNE     CON.A1                  ; PRINT UNTIL COUNT REACHES ZERO
012604  032704  000001                  BIT     #1,R4                   ; CHECK IF R4 NOW ODD
012610  001401                          BEQ     CON.A2                  ;
012612  0052C4                          INC     R4                      ; IF SO, INCREMENT TO NEXT EVEN ADDRESS
012614  000207                  CON.A2: RETURN                          ; NOW GET NEXT CONTROL CHARACTER

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

012616  012701  000012          CON.D:  MOV     #10.,R1                 ; LOAD RADIX
012622  004737  015552                  CALL    PNTNUM                  ; PRINT NUMBER
012626  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER

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

```
                                        ;
                                        ;
012630  012701  000020          CON.H:  MOV     #16.,R1                 ; LOAD RADIX
012634  004737  015552                  CALL    PNTNUM                  ; PRINT NUMBER
012640  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'O', SO PRINT AN OCTAL NUMBER.
                                        ;
012642  012701  000010          CON.O:  MOV     #8.,R1                  ; LOAD RADIX
012646  004737  015552                  CALL    PNTNUM                  ; PRINT NUMBER
012652  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'N', SO PRINT A CARRIAGE RETURN-LINE FEED.
                                        ;
012654  004737  015474          CON.N:  CALL    GETCNT                  ; GET COUNT
012660                          CON.N1: PRINT   #CR                     ;
012670  005301                          DEC     R1                      ; COUNT THE SEQUENCES
012672  001372                          BNE     CON.N1                  ;
012674  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'R', SO CALL ONE OF THE PRE-PROGRAMMED
                                        ;       ROUTINE.
                                        ;
012676  004737  015474          CON.R:  CALL    GETCNT                  ; GET ROUTINE NUMBER
012702  020127  000011                  CMP     R1,#ERR.SZ              ; CHECK IF DEFINED ROUTINE NUMBER
012706  101004                          BHI     CON.R1                  ;
012710  060101                          ADD     R1,R1                   ; DOUBLE COUNT TO GET WORD INDEX
012712  004771  012754                  CALL    @ERR.TB-2(R1)           ; CALL ROUTINE
012716  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
012720                          CON.R1: PNTF    ERRME1                  ; REPORT BAD MESSAGE STRING
012730                                  PCP     R1                      ; FIX THE STACK
012732  000207                          RETURN
                                        ;
                                        ;       CONTROL CHARACTER WAS AN 'S', SO PRINT SOME NUMBER OF SPACES.
                                        ;
012734  004737  015474          CON.S:  CALL    GETCNT                  ; GET COUNT
012740                          CON.S1: PRINT   <#' >                   ;
012750  005301                          DEC     R1                      ; COUNT THE SPACES
012752  001372                          BNE     CON.S1                  ;
012754  000207                          RETURN                          ; NOW GET NEXT CONTROL CHARACTER
                                        ;
                                        ;       PRE-PROGRAMMED ERROR ROUTINE DISPATCH TABLE
                                        ;
012756  015036                  ERR.TB: .WORD CALR1                     ; CALL ALTERNATE PRINT STRING IN DM PGM
012760  015064                          .WORD CALR2                     ; PRINT AN SDI DIAGNOSE RESPONSE
012762  015162                          .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 BASN1
                                        .WORD BASN2
                                        .WORD BASN3

                                ;
                                ;       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.O
                                        .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
                                        ;--

013040                                  TINIT:
013040  010137  002204                          MOV     R1,TNUM                 ; SAVE TEST NUMBER
013044  004737  020744                          CALL    RESET                   ; RESET / L CONTROLLERS
013050  013737  002164  002160                  MOV     FMEM,FFREE              ; INIT FREE
013056  013737  002166  002162                  MOV     FMEMS,FSIZE             ; INIT FSIZE
013064  006301                                  ASL     R1                      ; R1 IS WORD INDEX
013066  016137  013074  002200                  MOV     READDT-2(R1),DMPROG     ; STORE ADDRESS OF DM TEST INTO DMPROG
013074  000207                                  RETURN

013076                                  READDT:
013076  000000G                               .WORD   TEST1
013100  000000G                               .WORD   TEST2
013102  000000G                               .WORD   TEST3
```

```
                                ;**
                                ;       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
                                ;--

013104  010137  002206          RUNDM:  MOV     R1,URUN                 ; SAVE NUMBER OF UNITS TO RUN
013110  005037  002210                  CLR     URNING                  ; CLEAR NUMBER OF UNITS RUNNING
                                ;
                                ;       LOAD DM PROGRAM INTO EACH CONTROLLER
                                ;
013114  013705  002176                  MOV     TSTTAB,R5               ; GET CONTROLLER TABLE ADDRESS
013120  013737  002206  002212          MOV     URUN,UCNT               ; SET COUNTER OF UNITS
013126                          LDDM:
013126  005065  000012                  CLR     C.FLG(R5)               ; CLEAR ALL FLAGS
013132  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; SEE IF UNIT TO BE TESTED
013140  005765  000002                  TST     C.UNIT(R5)              ;
013144  100405                          BMI     LDNEXT                  ; IF NOT, DON'T LOAD THIS UNIT
013146                                  ASSUME CT.AVL EQ BIT15
013146  004737  016524                  CALL    LOADDM                  ; LOAD THE DM PROGRAM
013152  001402                          BEQ     LDNEXT                  ; IF ERROR, GO TO NEXT CONTROLLER
013154  005237  002210                  INC     URNING                  ; IF NO ERROR, COUNT UNIT RUNNING
013160  062705  000034          LDNEXT: ADD     #C.SIZE,R5              ; MOVE TO NEXT CONTROLLER TABLE
013164  005337  002212                  DEC     UCNT                    ; CHECK IF MORE CONTROLLERS
013170  001356                          BNE     LDDM                    ; LOAD NEXT CONTROLLER
                                ;
                                ;       CHECK IF ANY CONTROLLERS LOADED
                                ;
013172  005737  002210                  TST     URNING                  ; ANY UNITS LOADED?
                                ;
                                ;       THE DM PROGRAMS ARE NOW IN CONTROL
                                ;       RESPDM MUST BE CALLED TO RESPOND TO THEIR REQUESTS
                                ;
013176  000207                  RETURN                                  ;
```

```
                                        ;**
                                        ;       RESPDM - RESPOND TO DM REQUESTS. RETURN WHEN ALL DM PROGRAMS
                                        ;                HAVE TERMINATED.
                                        ;
                                        ;       INPUTS:
                                        ;                R5 - CONTROLLER TABLE ADDRESS
                                        ;--

013200  013705  002176          RESPDM: MOV     TSTTAB,R5               ; GET CONTROLLER TABLE ADDRESS
013204  013737  002206  002212          MOV     URUN,UCNT               ; SET COUNTER OF UNITS
013212  016504  000014          RESPCT: MOV     C.HCOM(R5),R4           ; GET HOST COMM AREA ADDRESS
013216  032765  000002  000012          BIT     #CT.RN,C.FLG(R5)        ; CHECK IF PROGRAM RUNNING
013224  001446                          BEQ     RSPNXT                  ;       IF NOT, LOOK AT NEXT
013226  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; STORE UNIT NUMBER UNDER TEST
013234  032765  000010  000012          BIT     #CT.MSG,C.FLG(R5)       ; SEE IF INTERRUPT RECEIVED
013242  001071                          BNE     RSPIN                   ;       IF SO, LOOK AT PACKET
013244  032765  000004  000012          BIT     #CT.CMD,C.FLG(R5)       ; SEE IF COMMAND HAS BEEN SENT
013252  001520                          BEQ     RSPOU                   ;       IF NOT, SEND ONE
                                        ;
                                        ;       CHECK IF CONTROLLER STILL RUNNING
                                        ;
013254  011503                          MOV     (R5),R3                 ; GET ADDRESS OF IP REGISTER
013256  016301  000002                  MOV     2(R3),R1                ; LOOK AT SA REGISTER
013262  001405                          BEQ     RSPTM                   ; IF ZERO, CONTROLLER STILL RUNNING
013264                                  ERRDF 30,,ERR030                ; ELSE, REPORT FATAL CONTROLLER ERROR
013264  104455                          TRAP    C$ERDF
013266  000036                          .WORD   30
013270  000000                          .WORD   0
013272  011430                          .WORD   ERR030
013274  000445                          BR      RSPDRP                  ; DROP CONTROLLER FROM TESTING
                                        ;
                                        ;       CHECK FOR TIMEOUT OF RESPONSE
                                        ;
013276                          RSPTM:
013276  005737  002256                  TST     KW.CSR                  ; SEE IF A CLOCK ON SYSTEM
013302  001416                          BEQ     RSPNTO                  ; DON'T TIME IF NO CLOCK
013304  023765  002270  000030          CMP     KW.EL+2,C.TOH(R5)       ; CHECK HIGH WORD OF ELAPSED TIME
013312  101005                          BHI     RSPTMO                  ; IF GREATER, RESPONSE TIMED OUT
013314  001011                          BNE     RSPNTO                  ; IF NOT SAME, ITS OK
013316  023765  002266  000026          CMP     KW.EL,C.TO(R5)          ; CHECK LOW WORD OF ELAPSED TIME
013324  103405                          BLO     RSPNTO                  ; IF LESS, PLENTY OF TIME LEFT
013326                          RSPTMO: ERRDF 31,,ERR031                ; REPORT TIMEOUT ERROR
013326  104455                          TRAP    C$ERDF
013330  000037                          .WORD   31
013332  000000                          .WORD   0
013334  011444                          .WORD   ERR031
013336  000424                          BR      RSPDRP                  ; DROP CONTROLLER FROM TESTING
013340                          RSPNTO:
013340                                  BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
013340  104422                          TRAP    C$BRK
                                        ;
                                        ;       CHECK FOR TIME TO PRINT STATISTICAL REPORT
                                        ;
013342  005737  002256          RSPNXT: TST     KW.CSR                  ; ANY CLOCK ON SYSTEM?
013346  001412                          BEQ     RSPNRP                  ;       BYPASS IF NOT
013350  023737  002270  002274          CMP     KW.EL+2,STIME+2         ; CHECK HIGH WORD OF ELAPSED TIME
```

```
     54 013356  101005                          BHI     RSPRPT                  ; IF GREATER PRINT REPORT
     55 013360  001005                          BNE     RSPNRP                  ; IF NOT SAME, ITS NOT TIME YET
     56 013362  023737  002266  002272          CMP     KW.EL,STIME             ; CHECK LOW WORD OF ELAPSED TIME
     57 013370  103401                          BLO     RSPNRP                  ; IF LESS, ITS NOT TIME YET
     58 013372                          RSPRPT:                                 ;
     59 013372                                  DORPT                           ; PRINT A STATISTICAL REPORT
        013372  104424                          TRAP    C$DRPT
     60                                 ;
     61                                 ;       SWITCH TO NEXT CONTROLLER
     62                                 ;
     63 013374  062705  000034          RSPNRP: ADD     #C.SIZE,R5              ; MOVE TO NEXT TABLE
     64 013400  005337  002212                  DEC     UCNT                    ; CHECK IF MORE CONTROLLERS
     65 013404  001302                          BNE     RESPCT                  ; LOOK AT NEXT CONTROLLER
     66 013406  000674                          BR      RESPDM                  ; LOOK AT FIRST CONTROLLER AGAIN
     67                                 ;
     68                                 ;       REMOVE A CONTROLLER FROM TESTING
     69                                 ;
     70 013410  042765  000012  000012  RSPDRP: BIC     #CT.RN+CT.MSG,C.FLG(R5) ; CLEAR PROGRAM RUNNING
     71 013416  005337  002210                  DEC     URNING                  ; REDUCE RUNNING CONTROLLERS COUNT
     72 013422  001347                          BNE     RSPNXT                  ; IF ANY STILL RUNNING, LOOK AT THEM
     73 013424  000207                          RETURN                          ; ELSE RETURN TO TEST SECTION
     74                                 ;
     75                                 ;       CONTROLLER HAS RESPONDED, LOOK AT MESSAGE PACKET
     76                                 ;       CHECK FOR PROPER OPCODE IN END PACKET
     77                                 ;
     78 013426  012700  000204          RSPIN:  MOV     #OP.END+OP.SSD,R0       ; GET SEND DATA END PACKET OPCODE
     79 013432  032765  000020  000012          BIT     #CT.REQ,C.FLG(R5)       ; LOOK IF SEND DATA OR RECEIVE DATA
     80 013440  001402                          BEQ     RSPMWR                  ;
     81 013442  012700  000205                  MOV     #OP.END+OP.RSD,R0       ; CHANGE TO RECEIVE DATA ENDCODE
     82 013446  120064  000030          RSPMWR: CMPB    R0,HC.MPK+P.OPCD(R4)    ; COMPARE TO OPCODE IN END PACKET
     83 013452  001010                          BNE     RSPERR                  ;
     84                                 ;
     85                                 ;       LOOK AT STATUS CODE
     86                                 ;
     87 013454  032764  000037  000032          BIT     #ST.MSK,HC.MPK+P.STS(R4); CHECK FOR STATUS CODE ST.SUC (ZERO)
     88 013462  001004                          BNE     RSPERR                  ;
     89                                 ;
     90                                 ;       CHECK FOR EXPECTED REFERENCE NUMBER
     91                                 ;
     92 013464  026564  000032  000020          CMP     C.REF(R5),HC.MPK+P.CRF(R4); CHECK IF CORRECT REF NUMBER
     93 013472  001405                          BEQ     RSPPTW                  ;
     94 013474                          RSPERR: ERPDF 33,,ERR033                ;
        013474  104455                          TRAP    C$ERDF
        013476  000041                          .WORD   33
        013500  000000                          .WORD   0
        013502  011474                          .WORD   ERR033
     95 013504  000741                          BR      RSPDRP                  ; DROP UNIT FROM TESTING
     96                                 ;
     97                                 ;       CHECK IF RESPONSE FROM SEND OR RECEIVE DATA COMMAND
     98                                 ;
     99 013506  032765  000020  000012  RSPPTW: BIT     #CT.REQ,C.FLG(R5)       ; CHECK IF RESPONSE FROM DM PROGRAM
    100 013514  001445                  RSPOU:  BEQ     RSPOUT                  ; LOOK AT REQUEST NUMBER IF SO
    101                                 ;
    102                                 ;       MAINTENANCE READ END PACKET RECEIVED, LOOK AT REQUEST FROM DM PROGRAM
    103                                 ;
    104 013516  016401  000206          RSPPT2: MOV     HC.BF2(R4),R1           ; GET REQUEST NUMBER
    105 013522  042701  007777                  BIC     #007777,R1              ; CHECK TYPE
```

```
                                                CMP     #DU.SPC,R1              ; IS SPECIAL TYPE SET?
                                                BNE     1$                      ; IF NOT, ERROR
                                                BIC     #^C007777,HC.OF2(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 DMRQ2                     ; 2 GET MANUAL INTERVENTION COMMAND
                                                .WORD 0                         ; 3 *** NOT VALID FOR CZUDHO ***
                                                .WORD 0                         ; 4 *** NOT VALID FOR CZUDHO ***
                                                .WORD 0                         ; 5 *** NOT VALID FOR CZUDHO ***
                                                .WORD 0                         ; 6 *** NOT VALID FOR CZUDHO ***
                                                .WORD DMRQ7                     ; 7 ADD TO SOFT ERROR AND ECC COUNTS
```

```
    160 013752  000000                .WORD 0                         ; 8 *** NOT VALID FOR CZUDHO ***
    161 013754  000000                .WORD 0                         ; 9 *** NOT VALID FOR CZUDHO ***
    162 013756  014564                .WORD DMRQA                     ;10 TELL WHICH DRIVES TO TEST
    163 013760  014626                .WORD DMRQB                     ;11 REPORT ERROR MESSAGE
    164 013762  014626                .WORD DMRQB                     ;[TEMP] 12 REPORT ERROR (FIX DM BUG)
    165 013764  014720                .WORD DMRQD                     ;13 PRINT A DESCRIPTIVE MESSAGE
    166 013766  015032                .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
                                ;--

013770                          DMRQ0:
013770  013701  002160                  MOV     FFREE,R1                ;GET FIRST ADDRESS OF FREE MEMORY
013774  013702  002162                  MOV     FSIZE,R2                ;GET SIZE
                                ;
                                ;       FILL MEMORY WITH ADDRESS PATTERN
                                ;
014000  010111                  MEMFIL: MOV     R1,(R1)                 ;WRITE DATA INTO LOCATION
014002  062701  000002                  ADD     #2,R1                   ;INCREASE ADDRESS TO NEXT LOCATION
014006  005302                          DEC     R2                      ;COUNT THE WORDS
014010  001373                          BNE     MEMFIL                  ;FILL ALL WORDS
                                ;
                                ;       SEND LOCATION OF FREE MEMORY TO CONTROLLER
                                ;
014012  013723  002160                  MOV     FFREE,(R3)+             ;LOAD FIRST ADDRESS OF FREE MEMORY
014016  005023                          CLR     (R3)+                   ; HIGH ORDER BITS ARE ZERO
014020  013700  002162                  MOV     FSIZE,R0                ;GET SIZE OF FREE MEMORY
014024  006300                          ASL     R0                      ;CONVERT TO BYTES
014026  063700  002160                  ADD     FFREE,R0                ;COMPUTE LAST LOCATION
014032  162700  000002                  SUB     #2,R0
014036  010023                          MOV     R0,(R3)+                ;LOAD LAST LOCATION
014040  005023                          CLR     (R3)+                   ;CLEAR HIGH ORDER BITS
                                ;
                                ;       SEND LOCATION OF READABLE MEMORY
                                ;
014042  005023                          CLR     (R3)+                   ;SEND ZERO AS START OF READABLE MEMORY
014044  005023                          CLR     (R3)+
014046  013700  002120                  MOV     L$HIMEM,R0              ;GET HIGH MEMORY ADDRESS
014052  005001                          CLR     R1                      ;CLEAR HIGH BITS
014054  006300                          ASL     R0                      ;SHIFT LEFT 6 PLACES
```

```
     58 014056  006300                    ASL     R0
     59 014060  006300                    ASL     R0
     60 014062  006300                    ASL     R0
     61 014064  006300                    ASL     R0
     62 014066  006101                    ROL     R1
     63 014070  006300                    ASL     R0
     64 014072  006101                    ROL     R1
     65 014074  052700  000076            BIS     #76,R0          ;SET LOW ORDER BITS
     66 014100  010023                    MOV     R0,(R3)+        ;PUT INTO BUFFER
     67 014102  010123                    MOV     R1,(R3)+
     68 014104  000264                    SEZ
     69 014106  000207                    RETURN
     70
```

```
;**
;       DMRQ2 - DM REQUEST 2
;
;       GET MANUAL INTERVENTION COMMAND
;
;       INPUTS:
;               R5 - CONTROLLER TABLE ADDRESS
;               R4 - MESSAGE DATA ADDRESS
;                  (R4) DRIVE NUMBER
;                2.(R4) OPERATION CODE
;                       0 ON FIRST REQUEST FOR DRIVE. ECHO OF PREVIOUS RESPONSE
;                       ALL OTHER TIMES.
;                 IF OPERATION CODE = 2
;                 4.(R4) DATA BYTE READ (TO BE PRINTED)
;               R3 - COMMAND DATA ADDRESS
;       OUTPUTS:
;               COMMAND DATA FILLED WITH THE FOLLOWING:
;                  (R3) OPERATION CODE
;                       0 - EXIT
;                       1 - WRITE
;                       2 - READ
;                       3 - DIAGNOSE
;                  IF OPERATION CODE = 1, 2 OR 3
;                 2.(R3) REGION NUMBER
;                 4.(R3) OFFSET INTO REGION
;                   IF OPERATION CODE = 1
;                 6.(R3) DATA BYTE
;               Z SET IF DATA RETURNED
;               Z CLEAR IF DRIVE NUMBER NOT ON THIS CONTROLLER
;--
DMRQ2:
        BIT     #SM.MAN,SFPTBL+SO.BIT   ;LOOK AT MANUAL INTERVENTION MODE
        BNE     DMRQ2M                  ;EXIT IF NOT WANTED
        JMP     DMRQ2X
DMRQ2M: MANUAL                          ;MANUAL INTERVENTION ALLOWED?
        TRAP    C$MANI
        BCOMPLETE DMRQ20                ;PRINT WARNING IF NOT
        BCS     DMRQ20
        PNTF    T2WARN
        JMP     DMRQ2X
DMRQ20: MOV     (R4)+,R1                ;GET DRIVE NUMBER
        MOV     (R4)+,R2                ;GET OPERATION CODE
        BNE     DMRQ22                  ;BRANCH IF NOT ZERO
        CALL    GTDRV1                  ;GET DRIVE TABLE ADDRESS
        BEQ     1$                      ;CHECK IF DRIVE FOUND
        RETURN                          ;RETURN WITH Z CLEAR IF NOT

1$:     PNTF    T2CMS1,D.UNIT(R4),(R5),(R4) ;PRINT DESCRIPTION
        CLR     T2WRR                       ;CLEAR ALL STORAGE WORDS
        CLR     T2WR0
        CLR     T2DR

DMRQ22: CMP     #2,R2                   ; SEE IF LAST OPERATION WAS READ
        BNE     DMRQ2Q                  ; BRANCH IF NOT TO QUESTION
        PRINT   <#' >                   ; PRINT ONE SPACE
        MOV     T2WRR,R1                ; PRINT REGION
```

GLOBAL SUBROUTINES SECTION

```
     56 014240  004737  016120                    CALL    T2PNTW                  ;
     57 014244  013701  002302                    MOV     T2WRO,R1                ; PRINT OFFSET
     58 014250  004737  016120                    CALL    T2PNTW                  ;
     59 014254                                    PRINT   #'/                     ; PRINT A SLASH
     60 014264  012401                            MOV     (R4)+,R1                ; PRINT THE DATA
     61 014266  004737  016150                    CALL    T2PNTB                  ;
     62 014272                                    PRINT   #CR                     ; END THE LINE
     63                                 ;
     64                                 ;       NOW ASK FOR COMMAND INPUT
     65                                 ;
     66 014302                          DMRQ2Q:  GMANID T4OPT7,TEMP,A,-1,1,20.,NO; ASK FOR COMMAND
        014302  104443                            TRAP    C$GMAN
        014304  000406                            BR      10000$
        014306  002216                            .WORD   TEMP
        014310  000142                            .WORD   T$CODE
        014312  002432                            .WORD   T4OPT7
        014314  177777                            .WORD   -1
        014316  000001                            .WORD   T$LOLIM
        014320  000024                            .WORD   T$HILIM
        014322                             10000$:
     67 014322  012701  002216                    MOV     #TEMP,R1                ; GET POINTER TO STRING
     68 014326  112100                            MOVB    (R1)+,R0                ; GET COMMAND CHARACTER
     69 014330  022700  000105                    CMP     #'E,R0                  ;
     70 014334  001415                            BEQ     DMRQ2V                  ;
     71 014336  022700  000104                    CMP     #'D,R0                  ;
     72 014342  001016                            BNE     DMRQ23                  ;
     73 014344  012713  000003                    MOV     #3,(R3)                 ; STORE DIAGNOSE OPERATION CODE
     74 014350  004737  016232                    CALL    T2GNUM                  ; GET REGION FROM COMMAND
     75 014354  001402                            BEQ     1$                      ;
     76 014356  010437  002304                    MOV     R4,T2DR                 ;
     77 014362  013763  002304  000002  1$:       MOV     T2DR,2(R3)              ;
     78 014370  004737  016232          DMRQ2V:  CALL    T2GNUM                  ; MAKE SURE AT END OF LINE
     79 014374  001064                            BNE     DMRQ2E                  ;
     80 014376  000461                            BR      DMRQ2X                  ;
     81                                 ;
     82                                 ;       COMMAND MUST BE EITHER READ OR WRITE
     83                                 ;
     84 014400  012713  000002          DMRQ23:  MOV     #2,(R3)                 ; CHECK IF READ
     85 014404  022700  000122                    CMP     #'R,R0                  ;
     86 014410  001415                            BEQ     DMRQ2R                  ;
     87 014412  022700  000127                    CMP     #'W,R0                  ; CHECK IF WRITE
     88 014416  001053                            BNE     DMRQ2E                  ;  IF NOT - ERROR
     89 014420  012713  000001                    MOV     #1,(R3)                 ;
     90 014424  004737  016232                    CALL    T2GNUM                  ; GET DATA BYTE
     91 014430  001446                            BEQ     DMRQ2E                  ; ERROR IF NO DATA
     92 014432  162700  000002                    SUB     #2,R0                   ;
     93 014436  003043                            BGT     DMRQ2E                  ; OR GREATER THAN TWO DIGITS
     94 014440  010463  000006                    MOV     R4,6(R3)                ; STORE DATA BYTES IN BUFFER
     95 014444  013763  002300  000002  DMRQ2R:  MOV     T2WRR,2(R3)             ; PUT REGION AND OFFSET
     96 014452  013763  002302  000004           MOV     T2WRO,4(R3)             ;      INTO BUFFER
     97 014460  021302                            CMP     (R3),R2                 ;
     98 014462  001002                            BNE     DMRQ2N                  ; IF SO,
     99 014464  005263  000004                    INC     4(R3)                   ;      INCREMENT OFFSET
    100 014470  004737  016232          DMRQ2N:  CALL    T2GNUM                  ; CONVERT ASCII TO NUMERIC
    101 014474  001411                            BEQ     DMRQ2W                  ;
    102 014476  010463  000002                    MOV     R4,2(R3)                ;
    103 014502  005063  000004                    CLR     4(R3)                   ;
```

```
    104 014506  004737  016232                      CALL    T2GNUM                  ; CONVERT ASCII TO NUMERIC
    105 014512  001402                              BEQ     DMRQ2W                  ;
    106 014514  010463  000004                      MOV     R4,4(R3)                ;
    107 014520  004737  016232          DMRQ2W:     CALL    T2GNUM                  ; CONVERT ASCII TO NUMERIC
    108 014524  001010                              BNE     DMRQ2E                  ;
    109 014526  016337  000002  002300              MOV     2(R3),T2WRR             ; SAVE REGION
    110 014534  016337  000004  002302              MOV     4(R3),T2WRO             ; SAVE OFFSET
    111 014542  000264                  DMRQ2X:     SEZ                             ; Z=0 FOR SUCCESSFUL COMPLETION
    112 014544  000207                              RETURN                          ;
    113 014546                          DMRQ2E:     PNTF    T2CMS5                  ; REPORT ERROR MESSAGE
    114 014556  000651                              BR      DMRQ2Q                  ; GO ASK AGAIN
    115
```

```
                                ;++
                                ;       DMRQ7 - DM REQUEST 7
                                ;
                                ;       THIS REQUEST IS USED ONLY TO KEEP THE HOST FROM TIMING OUT
                                ;       THE CONTOLLER WHILE DM TESTS 2 AND 3 ARE RUNNING.
                                ;--

014560  000264                  DMRQ7:  SEZ                             ; INDICATE SUCCESS
014562  000207                          RETURN                          ; EXIT
```

```
 1
 2                                      ;DMRQA - DM REQUEST 10
 3                                      ;
 4                                      ;TELL DM PROGRAM WHICH DRIVES ARE SELECTED FOR TESTING
 5                                      ;AND CLEAR STATISTICS IN DRIVE TABLE
 6                                      ;
 7                                      ;INPUTS:
 8                                      ;       R5 - CONTROLLER TABLE ADDRESS
 9                                      ;       R4 - MESSAGE DATA ADDRESS
10                                      ;            (NO DATA)
11                                      ;       R3 - COMMAND DATA ADDRESS
12                                      ;OUTPUTS:
13                                      ;       COMMAND PACKET CONTAINING UP TO 4 DRIVE NUMBERS.
14                                      ;         LIST IS ENDED BY A WORD WITH BIT 15 SET.
15                                      ;       D.XFRW, D.XFRR, D.HERR, D.SERR, D.SEEK AND D.ECC CLEARED
16                                      ;       Z SET
17
18 014564  010504                       DMRQA:  MOV     R5,R4                   ;GET ADDRESS OF CONTROLLER TABLE
19 014566  062704  000016                       ADD     #C.DRO,R4               ;BUMP TO DRIVE TABLE POINTERS
20 014572  012702  000004                       MOV     #4.,R2                  ;GET COUNT OF PORTS
21 014576  012400                       UTOT1:  MOV     (R4)+,R0                ;SEE IF DRIVE TABLE POINTER EXISTS
22 014600  001406                               BEQ     UTOT2                   ;BRANCH IF NOT
23 014602  005760  000002                       TST     D.UNIT(R0)              ;LOOK IF UNIT AVAILABLE FOR TESTING
24 014606  100401                               BMI     UTOT1A
25 014610                                       ASSUME DT.AVL EQ BIT15
26 014610  011023                               MOV     (R0),(R3)+              ;LOAD DRIVE NUMBER FROM TABLE
34 014612  005302                       UTOT1A: DEC     R2                      ; COUNT THE DRIVE TABLES
35 014614  001370                               BNE     UTOT1                   ; REPEAT FOR EACH TABLE
36 014616  012723  100000               UTOT2:  MOV     #BIT15,(R3)+            ; TERMINATE LIST
37 014622  000264                               SEZ                             ;
38 014624  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
014626                          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$:   POP     R4                      ;RESTORE R4
                                ;;
014626  005764  000002          1$:     TST     2(R4)                   ;CHECK IF FATAL ERROR
014632  100406                          BMI     5$                      ;BRANCH IF NOT
014634                                  RFLAGS  R0                      ;LOOK AT FLAGS
014634  104421                          TRAP    C$RFLA
014636  032700  000040                  BIT     #IDU,R0                 ;SEE IF ALLOWED TO DROP UNITS
014642  001014                          BNE     6$                      ;BRANCH IF NOT
014644  052713  100000                  BIS     #BIT15,(R3)             ;SET DROP DRIVE BIT
014650  016400  000002          5$:     MOV     2(R4),R0                ;SEE IF SOFT ERROR
014654  005100                          COM     R0
014656  032700  140000                  BIT     #140000,R0
014662  001004                          BNE     6$                      ;BRANCH IF NOT
014664  032737  000400  002146          BIT     #SM.SSF,SO.BIT+SFPTBL   ;SEE IF SOFT ERRORS SUPPRESSED
014672  001010                          BNE     ERRMSX                  ;DON'T PRINT IF SO
014674                          6$:
014674  042765  000010  000012          BIC     #CT.MSG,C.FLG(R5)       ;CLEAR MESSAGE RECEIVED FLAG
014702  004737  016360                  CALL    PNTERR                  ; PRINT ERROR MESSAGE
014706  103002                          BCC     ERRMSX                  ; IF DRIVE HASN'T BEEN DROPPED, PRINT
014710                          8$:
014710  000244                          CLZ                             ;       ELSE RETURN
014712  000207                          RETURN
```

```
61
83 014714  000264                ERRMSX: SEZ
84 014716  000207                        RETURN
85
```

```
                                ;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

014720  042765  000010  000012  DMRQD:  BIC     #CT.MSG,C.FLG(R5)       ;CLEAR MESSAGE RECEIVED FLAG
014726  012401                          MOV     (R4)+,R1                ;GET DRIVE NUMBER
014730                                  PUSH    R4                      ;SAVE DATA POINTER
014732  004737  015404                  CALL    GTDRVT                  ;GET DRIVE TABLE ADDRESS
014736  001033                          BNE     1$                      ;CHECK IF DRIVE FOUND
014740  005764  000002                  TST     D.UNIT(R4)              ;IF UNIT DROPPED FROM TESTING
014744  100430                          BMI     1$                      ;    DON'T PRINT ANYTHING
014746                                  PNTX    MESSG,D.UNIT(R4),(R5),(R4); PRINT HEADER
014766  004737  021046                  CALL    RNTIME                  ; GET RUNTIME PARAMETERS
014772                                  POP     R4                      ; RESTORE MESSAGE MOINTER
014774  012402                          MOV     (R4)+,R2                ;GET MESSAGE POINTER
014776  006302                          ASL     R2                      ;DOUBLE TO MAKE BYTE OFFSET
015000  063702  002200                  ADD     DMPROG,R2               ;ADD TO START OF MESSAGE STRINGS
015004  067702  165170                  ADD     @DMPROG,R2              ;ADD SIZE OF MAIN PROGRAM
015010  105712                          TSTB    (R2)                    ;CHECK FIRST BYTE
015012  001001                          BNE     2$                      ;IF ZERO
015014  005202                          INC     R2                      ;  INCREMENT TO NEXT BYTE
015016  004737  012476          2$:     CALL    OSTRNG                  ;OUTPUT ACCORDING TO STRING
015022  000264                          SEZ
015024  000207                          RETURN
015026                          1$:     POP     R4
015030  000207                          RETURN
```

```
      1
      2                                 ;DMRQE - DM REQUEST 14
      3                                 ;
      4                                 ;MARK DM PROGRAM AS NO LONGER RUNNING
      5                                 ;
      6                                 ;INPUTS:
      7                                 ;       R5 - CONTROLLER TABLE ADDRESS
      8                                 ;       R4 - MESSAGE DATA ADDRESS
      9                                 ;            (NO DATA)
     10                                 ;       R3 - COMMAND DATA ADDRESS
     11                                 ;OUTPUTS:
     12                                 ;       Z CLEAR TO DROP UNIT FROM TESTING
     13
     14 015032  000244                  DMRQE:  CLZ                             ;DROP UNIT FROM TESTING
     15 015034  000207                          RETURN
```

```
                                .SBTTL  PRE-PROGRAMMED SUBROUTINES
                                ;++
                                ;       CALR1 - PRE-PROGRAMMED PRINT ROUTINE 1
                                ;
                                ;       CALL ALTERNATE PRINT STRING IN DM PROGRAM IMAGE
                                ;--
                                CALR1:  PUSH    R2
015036                                  MOV     (R4)+,R2                ;GET NEW STRING POINTER
015040  012402                          ASL     R2                      ;DOUBLE FOR WORD COUNT
015042  006302                          ADD     DMPROG,R2               ;ADD START OF STRING STORAGE
015044  063702  002200                   ADD     @DMPROG,R2              ;ADD SIZE OF MAIN PROGRAM
015050  067702  165124                   CALL    OSTRNG                  ;OUTPUT USING THIS STRING
015054  004737  012476                   POP     R2                      ;GET OLD POINTER BACK
015060                                  RETURN                          ;NOW CONTINUE THE OLD STRING
015062  000207
                                ;++
                                ;       CALR2 - PRE-PROGRAMMED PRINT ROUTINE 2
                                ;
                                ;       PRINT AN SDI DIAGNOSE RESPONSE
                                ;--
                                CALR2:  PUSH    R2
015064                                  MOV     (R4)+,R2                ;GET COUNTS
015066  012402                          PUSH    R2                      ;SAVE COUNTS
015070                                  BIC     #177400,R2              ;GET BINARY COUNT
015072  042702  177400                   BEQ     2$                      ;BYPASS BINARY IF COUNT IS ZERO
015076  001414                  1$:     MOV     #16.,R0                 ;RADIX IS HEX
015100  012700  000020                   MOV     #32.,R1                 ;32 BIT NUMBERS
015104  012701  000040                   CALL    PNTNUS                  ;PRINT THE NUMBER
015110  004737  015560                   PRINT   #CR                     ;GO TO NEW LINE
015114                                  DEC     R2
015124  005302                          BNE     1$
015126  001364                  2$:     POP     R1
015130                                  SWAB R1                         ;GET ASCII COUNT
015132  000301                          BIC     #177400,R1
015134  042701  177400                   BEQ     3$                      ;BYPASS IS COUNT IS ZERO
015140  001406                          CALL    CON.A1                  ;PRINT THE ASCII
015142  004737  012572                   PRINT   #CR                     ;GO TO NEW LINE
015146                                  POP     R2
015156                          3$:     RETURN
015160  000207
                                ;++
                                ;       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
                                ;--
                                CALR3:  BIT     #BIT7,(R4)+             ;CHECK BIT 7 IN FIRST PARAMETER WORD
015162  032724  000200                  BNE     CALR1                   ;IF SET, TURN INTO A CALR1
015166  001323
```

```
     58 015170  062704  000006                ADD     #6,R4                   ;ELSE, SKIP OVER NEXT 3 PARAMETERS
     59 015174  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 015176                        CALR4:  PNTB    BASLN,#BASNO,#BAS,#BAS,#BAS
     69 015226  004737  021046                CALL    RNTIME
     70 015232                                PRINT   #CR
     71 015242  012737  012322  002306        MOV     #PX,PTYPE
     72 015250  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 015252                        CALR5:  PNTB    BASLN,#BASNO,#BASL2,(R5),#BAS,#BAS
     82 015304  004737  021046                CALL    RNTIME
     83 015310                                PRINT   #CR
     84 015320  012737  012322  002306        MOV     #PX,PTYPE
     85 015326  000207                        RETURN
     86
     87                                ;++
     88                                ;       CALR6 - PRE-PROGRAMMED PRINT ROUTINE 6
     89                                ;
     90                                ;       CALL ALTERNATE PRINT ROUTINE IN PDP-11 MEMORY
     91                                ;--
     92
     93 015330                        CALR6:  PUSH    R2
     94 015332  012402                        MOV     (R4)+,R2                ;GET NEW STRING POINTER
     95 015334  004737  012476                CALL    OSTRNG                  ;OUTPUT USING THIS STRING
     96 015340                                POP     R2
     97 015342  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 015344                        CALR7:  PUSH    R2
    106 015346  012702  010672                MOV     #XFRU,R2
    107 015352  004737  012476                CALL    OSTRNG
    108 015356                                POP     R2
    109 015360  000207                        RETURN
    110
    111                                ;++
    112                                ;       CALR8 - PRE-PROGRAMMED PRINT ROUTINE 8
    113                                ;
    114                                ;       PRINT   "  SA REGISTER CONTAINS  XXXXXX"
```

```
                                        ;--
                                        ;
015362                          CALR8:  PUSH    R2
015364  012702  010635                  MOV     #XSA,R2
015370  004737  012476                  CALL    OSTRNG
015374                                  POP     R2
015376  000207                          RETURN
                                        ;
                                        ;++
                                        ;       CALR9 - PRE-PROGRAMMED PRINT ROUTINE 9
                                        ;
                                        ;       REPRINT LAST NUMBER
                                        ;               R4 -> TABLE
                                        ;--
                                        ;
015400  005744                  CALR9:  TST     -(R4)
015402  000207                          RETURN
                                        ;
```

```
 1
 2                                          ;GTDRVT
 3                                          ;
 4                                          ;GET DRIVE TABLE POINTER
 5                                          ;
 6                                          ;INPUTS:
 7                                          ;       R5 - CONTROLLER TABLE ADDRESS
 8                                          ;       R1 - DRIVE NUMBER
 9                                          ;OUTPUTS:
10                                          ;       R4 - DRIVE TABLE ADDRESS
11                                          ;       L$LUN - LOADED WITH UNIT NUMBER OF DRIVE
12                                          ;       Z CLEAR IF DRIVE TABLE NOT FOUND AFTER ERROR PRINTED
13
14 015404                                  GTDRVT: PUSH    R2
15 015406  010504                                  MOV     R5,R4                   ;GET CONTROLLER TABLE ADRESS
16 015410  062704  000016                          ADD     #C.DRO,R4               ;ADD OFFSET TO DRIVE TABLE ADDRESS
17 015414  012702  000004                          MOV     #4.,R2                  ;GET COUNT OF DRIVES
18 015420  005714                          1$:     TST     (R4)                    ;CHECK IF AN ADDRESS HERE
19 015422  001406                                  BEQ     3$
20 015424  027401  000000                          CMP     @(R4),R1                ;COMPARE DRIVE NUMBERS
21 015430  001412                                  BEQ     4$                      ;BRANCH IF A MATCH
22 015432  005724                          2$:     TST     (R4)+                   ;BUMP ADDRESS
23 015434  005302                                  DEC     R2
24 015436  001370                                  BNE     1$                      ;LOOK AT ALL OF THEM
25 015440                                  3$:     ERRDF 35.,ERR035                ;UNIT NUMBER NOT FOUND
   015440  104455                                  TRAP    C$ERDF
   015442  000043                                  .WORD   35
   015444  000000                                  .WORD   0
   015446  011502                                  .WORD   ERR035
26 015450                                          POP     R2
27 015452  000244                                  CLZ                             ;CLEAR Z AS ERROR FLAG
28 015454  000207                                  RETURN
29
30 015456  011404                          4$:     MOV     (R4),R4                 ;GET ADDRESS OF TABLE
31 015460  116437  000002  002074                  MOVB    D.UNIT(R4),L$LUN        ;GET UNIT NUMBER
32 015466                                          POP     R2
33 015470  000264                                  SEZ                             ;SET Z FLAG
34 015472  000207                                  RETURN
```

```
 1
 2                                      ;GETCNT
 3                                      ;
 4                                      ;GET COUNT IN NEXT CHARACTERS OF STRING POINTED TO BY R2.
 5                                      ;NUMBER WILL BE IN DECIMAL. IF NO NUMBER, RETURN A
 6                                      ;DEFAULT OF 1.
 7                                      ;
 8                                      ;INPUTS:
 9                                      ;       R2 - POINTER TO ASCII STRING
10                                      ;OUTPUTS:
11                                      ;       R1 - NUMBER READ OR A ONE
12                                      ;       R2 - POINTING TO CHARACTER AFTER NUMBER
13
14 015474                              GETCNT: PUSH    R0
15 015476  005001                              CLR     R1                      ;START WITH ZERO COUNT
16 015500  121227  000060           GETCNX:     CMPB    (R2),#'0                ;CHECK IF CHARACTER A DIGIT
17 015504  103415                              BLO     GETCDN                  ;BRANCH IF LOWER THAN ZERO
18 015506  121227  000071                      CMPB    (R2),#'9
19 015512  101012                              BHI     GETCDN                  ;BRANCH IF HIGHER THAN NINE
20 015514  006301                              ASL     R1                      ;MULTIPLY NUMBER BY 10
21 015516  010100                              MOV     R1,R0                   ; SAVE 2N
22 015520  006301                              ASL     R1                      ; COMPUTE 4N
23 015522  006301                              ASL     R1                      ; COMPUTE 8N
24 015524  060001                              ADD     R0,R1                   ; 8N + 2N = 10N
25 015526  112200                              MOVB    (R2)+,R0                ;GET DIGIT FROM STING
26 015530  162700  000060                      SUB     #'0,R0                  ;GET RID OF ASCII
27 015534  060001                              ADD     R0,R1                   ;ADD TO NUMBER
28 015536  000760                              BR      GETCNX                  ;GO TO NEXT CHARACTER
29 015540  005701                  GETCDN:     TST     R1                      ;CHECK IF NUMBER IS ZERO
30 015542  001001                              BNE     GETCXX                  ;IF ZERO, CHANGE
31 015544  005201                              INC     R1                      ; TO DEFAULT OF ONE
32 015546                          GETCXX:     POP     R0
33 015550  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
                                        ;
015552  010100          PNTNUM: MOV     R1,R0           ; SAVE RADIX
015554  004737  015474          CALL    GETCNT          ; GET COUNT OF BITS
015560                  PNTNUS. PUSH    <R2,R3,R5>      ; SAVE REGISTERS ON STACK
015566  012403                  MOV     (R4)+,R3        ; GET ONE PARAMETER WORD
015570  005005                  CLR     R5              ; CLEAR STORAGE FOR OTHER
015572  020127  000020          CMP     R1,#16.         ; MORE THAN 16 BITS IN NUMBER?
015576  003401                  BLE     1$              ; NO, SKIP
015600  012405                  MOV     (R4)+,R5        ; YES, GET SECOND PARAMETER WORD
015602                  1$:     PUSH    R4              ; SAVE R4 ON STACK
015604  010504                  MOV     R5,R4           ; PUT LOW WORD IN R4
015606  012702  000020          MOV     #16.,R2         ; COMPUTE BITS NOT WANTED
015612  160102                  SUB     R1,R2           ; BY SUBTRACTING BITS TO USE
015614  002002                  BGE     2$              ; FROM 16.
015616  062702  000020          ADD     #16.,R2         ; IF NEGATIVE, ADD 16 FOR FIRST WORD
015622  001414          2$:     BEQ     6$              ; IF ZERO, NO BITS NEED BE CLEARED
015624  012705  100000          MOV     #BIT15,R5       ; START MASK WITH SIGN BIT SET
015630  005302          3$:     DEC     R2              ; COUNT BITS IN MASK
015632  001402                  BEQ     4$              ;
015634  006205                  ASR     R5              ; SHIFT MORE BITS TO RIGHT
015636  000774                  BR      3$              ;
015640  020127  000020  4$:     CMP     R1,#16.         ; MORE THAN 16 BITS IN NUMBER?
015644  001402                  BLE     5$              ;
015646  040504                  BIC     R5,R4           ; YES, CLEAR IN HIGH WORD
015650  000401                  BR      6$              ;
015652  040503          5$:     BIC     R5,R3           ; NO, CLEAR IN LOW WORD
015654  004737  016014  6$:     CALL    DIVIDE          ; DIVIDE BY RADIX IN R0
015660                          PUSH    R5              ; PUSH REMAINDER ON STACK
015662  005202                  INC     R2              ; COUNT DIGITS ON STACK
015664  005703                  TST     R3              ; CHECK IF QUOTIENT IS ZERO
015666  001372                  BNE     6$              ;
015670  005704                  TST     R4              ;
015672  001370                  BNE     6$              ;
015674  020027  000012          CMP     R0,#10.         ; IF RADIX IS DECIMAL
015700  001423                  BEQ     10$             ;       JUST GO PRINT DIGITS ON STACK
015702  010103                  MOV     R1,R3           ; OTHERWISE COMPUTE NUMBER OF LEADING 0
015704  162700  000014          SUB     #12.,R0         ; DIVIDEND IS BITS IN NUMBER
015710  003002                  BGT     7$              ; DIVISOR IS BITS PER DIGIT PRINTED
015712  012700  000003          MOV     #3,R0           ;       (3 OR 4)
015716  004737  016014  7$:     CALL    DIVIDE          ;
015722  005705                  TST     R5              ; IF REMAINDER NOT ZERO
015724  001401                  BEQ     8$              ; INCREMENT QUOTIENT
015726  005203                  INC     R3              ;
015730  160203          8$:     SUB     R2,R3           ; SUBTRACT DIGITS ON STACK
```

```
      58 015732  001406                       BEQ     10$                     ; NO LEADING ZEROS IF ZERO
      59 015734                          9$:   PRINT   #'0                     ;
      60 015744  005303                       DEC     R3                      ;
      61 015746  001372                       BNE     9$                      ; REPEAT UNTIL COUNT REACHES ZERO
      62 015750                          10$:  POP     R5                      ; GET CHACACTER FROM STACK
      63 015752  062705  000060               ADD     #'0,R5                  ; CNVERT TO ASCII DIGIT
      64 015756  020527  000071               CMP     R5,#'9                  ; IF GREATER THAN A 9
      65 015762  003402                       BLE     11$                     ;       CONVERT TO A OR HIGHER
      66 015764  062705  000007               ADD     #<'A-'9-1>,R5           ;       FOR HEX DIGIT
      67 015770                          11$:  PRINT   R5                      ;
      68 015776  005302                       DEC     R2                      ; REPEAT FOR ALL DIGITS
      69 016000  001363                       BNE     10$                     ;       ON STACK
      70 016002                               POP     <R4,R5,R3,R2>           ;
      71 016012  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

016014                                  DIVIDE: PUSH    R2
016016  012702  000040                          MOV     #32.,R2         ;SET UP SHIFT COUNT
016022  005005                          1$:     CLR     R5              ;START WITH ZERO REMAINDER
016024  006303                          1$:     ASL     R3              ;SHIFT LEFT INTO R5
016026  006104                                  ROL     R4
016030  006105                                  ROL     R5
016032  020005                                  CMP     R0,R5           ;WILL DIVISOR GO INTO REMAINDER
016034  101002                                  BHI     2$              ;ONLY SUBTRACT IF IT WILL
016036  160005                                  SUB     R0,R5           ;SUBTRACT DIVISOR
016040  005203                                  INC     R3              ;PUT A ONE INTO QUOTIENT
016042  005302                          2$:     DEC     R2              ;COUNT THE SHIFTS
016044  001367                                  BNE     1$
016046                                          POP     R2
016050  000207                                  RETURN
```

```
                                ;DIV10
                                ;
                                ;DIVIDE A 64 BIT UNSIGNED NUMBER BY A 10.
                                ;REPLACE DIVIDEND WITH QUOTIENT AND RETURN REMAINDER.
                                ;WILL NOT CHECK FOR DIVIDE BY ZERO.
                                ;
                                ;INPUTS:
                                ;       R1 - LOW  16 BITS OF DIVIDEND
                                ;       R2 - NEXT 16 BITS OF DIVIDEND
                                ;       R3 - NEXT 16 BITS OF DIVIDEND
                                ;       R4 - HIGH 16 BITS OF DIVIDEND
                                ;OUTPUTS:
                                ;       R1 - QUOTIENT,
                                ;       R2 - QUOTIENT,
                                ;       R3 - QUOTIENT,
                                ;       R4 - QUOTIENT,
                                ;       R5 - REMAINDER
                                
016052                  DIV10:  PUSH    R0                      ;DIVIDEND IS IN <R4,R3,R2,R1>
016054  012700  000100          MOV     #64.,R0                 ;SET UP SHIFT COUNT
016060  005005                  CLR     R5                      ;START WITH ZERO REMAINDER
016062  006301          1$:     ASL     R1
016064  006102                  ROL     R2                      ;SHIFT LEFT INTO R5
016066  006103                  ROL     R3
016070  006104                  ROL     R4
016072  006105                  ROL     R5
016074  022705  000012          CMP     #10.,R5                 ;SILL DIVISOR GO INTO REMAINDER?
016100  101003                  BHI     2$                      ;ONLY SUBTRACT IF IT WILL
016102  162705  000012          SUB     #10.,R5                 ;SUBTRACT DIVISOR
016106  005201                  INC     R1                      ;PUT A ONE INTO QUOTIENT
016110  005300          2$:     DEC     R0                      ;COUNT THE SHIFTS
016112  001363                  BNE     1$
016114                          POP     R0                      ;RETURN WITH QUOTIENT IN
016116  000207                  RETURN                          ;  <R4,R3,R2,R1> AND REMAINDER IN R5
```

```
 1
 2                                      ;++
 3                                      ;       T2PNTW - PRINT HEX NUMBERS WITH LEADING SPACE
 4                                      ;
 5                                      ;       THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
 6                                      ;--
 7
 8 016120                               T2PNTW: PRINT   <#' >                   ; PRINT A SPACE
 9 016130                                       PUSH    R1                      ; SAVE R1 ON STACK
10 016132  000301                               SWAB    R1                      ; GET HIGH BYTE
11 016134  004737  016160                       CALL    T2PNT                   ; PRINT HIGH TWO DIGITS
12 016140                                       POP     R1                      ; RESTORE R1
13 016142  004737  016160                       CALL    T2PNT                   ; PRINT LOW TWO DIGITS
14 016146  000207                               RETURN                          ;
15
16                                      ;++
17                                      ;       T2PNTB - PRINT A SPACE
18                                      ;
19                                      ;       THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTEPVENTION QUESTIONS
20                                      ;--
21
22 016150                               T2PNTB: PRINT   <#' >                   ; PRINT A SPACE
23
24                                      ;++
25                                      ;       T2PNT - PRINT TWO HEX DIGITS FROM NUMBER IN R1
26                                      ;
27                                      ;       THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
28                                      ;--
29
30 016160                               T2PNT:  PUSH    R1                      ; SAVE NUMBER
31 016162  006001                               ROR     R1                      ; SHIFT
32 016164  006001                               ROR     R1                      ;      TO GET
33 016166  006001                               ROR     R1                      ;            HIGH
34 016170  006001                               ROR     R1                      ;                 DIGIT
35 016172  004737  016200                       CALL    T2PNTO                  ; PRINT TWO DIGITS
36 016176                                       POP     R1                      ; GET LOW DIGIT AGAIN
37 016200  042701  177760              T2PNTO: BIC     #^C17,R1                ; CLEAR OTHER BITS
38 016204  062701  000060                       ADD     #'0,R1                  ; CONVERT TO ASCII CHARACTER
39 016210  020127  000071                       CMP     R1,#'9                  ; IF GREATER THAN 9
40 016214  003402                               BLE     T2PNTD                  ;   CONVERT
41 016216  062701  000007                       ADD     #<'A-'9-1>,R1           ;          TO A - F
42 016222                              T2PNTD: PRINT   R1                      ; PRINT THE DIGIT
43 016230  000207                               RETURN
44
```

```
                                ;**
                                ;       T2GNUM - GET A HEX DIGIT FROM AN ASCII INPUT STRING
                                ;
                                ;       THIS ROUTINE IS USED ONLY FOR TEST 2 MANUAL INTERVENTION QUESTIONS
                                ;
                                ;       INPUTS:
                                ;               R1 - STRING POINTER
                                ;       OUTPUTS:
                                ;               R4 - NUMBER
                                ;               R1 - UPDATED STRING TO CHARACTER AFTER NUMBER
                                ;               R0 - COUNT OF DIGITS (0 IF END OF LINE FOUND)
                                ;--

C16232  005000          T2GNUM: CLR     R0              ; CLEAR DIGIT COUNT
016234  105711                  TSTB    (R1)            ; CHECK IF END OF LINE
016236  001442                  BEQ     T2GNX           ; REPORT NULL CHARACTER FOUND
016240  121127  000040          CMPB    (R1),#' '       ; CHECK IF A SPACE
016244  001002                  BNE     T2GND1          ; IF SO, IGNORE IT
016246  005201                  INC     R1              ; GET NEXT
016250  000770                  BR      T2GNUM          ;       DIGIT
016252  005004          T2GND1: CLR     R4              ; CLEAR NUMBER STORAGE
016254          T2GND2: PUSH    R2                      ; SAVE REGISTER
016256  112102                  MOVB    (R1)+,R2        ; GET CHARACTER
016260  162702  000060          SUB     #'0,R2          ; CONVERT ASCII TO NUMERIC
016264  100431                  BMI     T2GNE           ; IF LESS THAN 0, INVALID DIGIT
016266  020227  000011          CMP     R2,#9.          ; IF LESS THAN OR EQUAL TO 9
016272  003410                  BLE     T2GND3          ;       CONVERSION DONE
016274  020227  000021          CMP     R2,#<'A-'0>     ; IF LESS THAN A (10.),
016300  103423                  BLO     T2GNE           ;       INVALID DIGIT
016302  020227  000026          CMP     R2,#<'F-'0>     ; IF GREATER THAN F (15.),
016306  101020                  BHI     T2GNE           ;       INVALID DIGIT
016310  162702  000007          SUB     #<'A-'9-1>,R2   ; CONVERT FROM A-F TO 10.-15.
016314  006304          T2GND3: ASL     R4              ; MAKE
016316  006304                  ASL     R4              ;    ROOM
016320  006304                  ASL     R4              ;       FOR
016322  006304                  ASL     R4              ;         DIGIT
016324  050204                  BIS     R2,R4           ; STORE DIGIT
016326  005200                  INC     R0              ; GET NEXT
016330                          POP     R2              ;       DIGIT
016332  105711                  TSTB    (R1)            ; IF 0,
016334  001403                  BEQ     T2GNX           ;    END OF NUMBER
016336  121127  000040          CMPB    (R1),#' '       ; IF NOT A SPACE,
016342  001344                  BNE     T2GND2          ;    CONTINUE
016344  005700          T2GNX:  TST     R0              ; Z = 1 FOR ERROR, 0 FOR SUCCESS
016346  000207                  RETURN                  ;

016350          T2GNE:  POP     <R2,R0>         ; INVALID DIGIT
016354  000137  014546          JMP     DMRQ2E          ;       ERROR
```

```
                                        ;++
                                        ;       PNTERR - PRINT ERROR MESSAGE FROM DM PROGRAM REQUEST #1 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
                                        ;--

016360                          PNTERR: PUSH    <R0,R1,R2>              ; SAVE REGISTERS ON STACK
016366  005764  000004                  TST     4(R4)                   ; CHECK IF BIT 15 SET
016372  002004                          BGE     1$                      ; IF SO, GET UNIT FROM MESSAGE DATA
016374  116537  000002  002074          MOVB    C.UNIT(R5),L$LUN        ; ELSE, GET UNIT FROM CONTROLLER TABLE
016402  000417                          BR      2$                      ;
016404                          1$:     PUSH    R4                      ; SAVE DATA ADDRESS
016406  016401  000004                  MOV     4(R4),R1                ; GET DRIVE NUMBER
016412  004737  015404                  CALL    GTDRVT                  ; GET DRIVE TABLE ADDRESS
016416  001037                          BNE     5$                      ; IF UNIT DROPPED, EXIT
016420  005764  000002                  TST     D.UNIT(R4)              ; IS TESTING DONE ON UNIT?
016424  100005                          BPL     3$                      ; IF NOT DONE, PROCEED
016426  052713  100000                  BIS     #BIT15,(R3)             ; MARK UNIT AS DONE TESTING
016432                                  DORPT                           ; PRINT A STATISTICAL REPORT
016432  104424                          TRAP    C$DRPT
016434                                  POP     R4                      ; RESTORE DATA ADDRESS
016436  000423                          BR      4$                      ;

016440                          3$:     POP     R4                      ; RESTORE DATA ADDRESS
016442  012702  002150          2$:     MOV     #ERRTYP,R2              ; GET POINTER TO ERROR TABLE
016446  016412  000002                  MOV     2(R4),(R2)              ; GET ERROR TYPE
016452  006112                          ROL     (R2)                    ;
016454  006112                          ROL     (R2)                    ;
016456  006112                          ROL     (R2)                    ;
016460  042722  177774                  BIC     #^C3,(R2)+              ; CLEAR LOW 2 BITS
016464  016412  000002                  MOV     2(R4),(R2)              ;
016470  042722  140000                  BIC     #140000,(R2)+           ; MASK LOW 14 BITS
016474  005022                          CLR     (R2)+                   ; CLEAR MESSAGE POINTER
016476  012712  011760                  MOV     #ERR.TN,(R2)            ; GET ROUTINE NUMBER
016502                                  ERROR                           ; PRINT THE ERROR MESSAGE
016502  104460                          TRAP    C$ERROR
016504  000241                          CLC                             ; DRIVE HAS NOT BEEN DROPPED
016506                          4$:     POP     <R2,R1,R0>              ; RESTORE REGISTERS
016514  000207                          RETURN                          ;
016516                          5$:     POP     <R4>                    ; RESTORE STACK
016520  000261                          SEC                             ; DRIVE HAS BEEN DROPPED
016522  000771                          BR      4$                      ;
```

```
                                        ;++
                                        ;       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
                                        ;--

016524                                  LOADDM:
016524  016504  000004                          MOV     C.VEC(R5),R4            ; GET VECTOR OF CONTROLLER
016530  042704  177000                          BIC     #^C<CT.VEC>,R4          ; CLEAR ALL BUT VECTOR
016534  010501                          MOV     R5,R1                   ; GET INTERRUPT SERVICE LINK
016536  062701  000006                  ADD     #C.JSR,R1               ;
016542                                  SETVEC  R4,R1,#PRIO7            ; SET UP INTERRUPT VECTOR
016542  012746  000340                  MOV     #PRIO7,-(SP)
016546  010146                          MOV     R1,-(SP)
016550  010446                          MOV     R4,-(SP)
016552  012746  000003                  MOV     #3,-(SP)
016556  104437                          TRAP    C$SVEC
016560  062706  000010                  ADD     #10,SP
016564  006204                          ASR     R4                      ; INITIALIZE CONTROLLER WITH SMALLEST
016566  006204                          ASR     R4                      ;       POSITION VECTOR FOR RING
016570  004737  017676                  CALL    CNTINT                  ;       BUFFER AND INTERRUPTS ENABLED
016574  001526                          BEQ     LOADER                  ; IF ERROR, EXIT
016576  004737  012156                  CALL    HCOMM                   ; ALLOCATE SPACE FOR HOST COMM AREA
016602  023727  002204  000001          CMP     TNUM,#1                 ; IF TEST NUMBER <> 1
016610  001053                          BNE     LOADTX                  ;       DON'T DO SPECIAL LOAD
                                        ;
                                        ;       LOAD DM PROGRAM INTO MEMORY SPACE TESTED DURING
                                        ;       INITIALIZATION IN TEST 1
                                        ;
016612  017704  163362                  MOV     @DMPROG,R4              ;GET SIZE OF DM PROGRAM IN BYTES
016616  162704  000040                  SUB     #DMMAIN,R4              ;
016622  013700  002200                  MOV     DMPROG,R0               ;GET ADDRESS OF DM PROGRAM
016626  062700  000040                  ADD     #DMMAIN,R0              ;
016632  005001                          CLR     R1                      ;START WITH OFFSET OF ZERO

016634  012703  000214          LT1L1:  MOV     #<HC.BSZ+2>,R3          ; GET SIZE OF BUFFER
016640  020403                          CMP     R4,R3                   ; IF FEWER BYTES REMAINING IN PROGRAM
016642  103001                          BHIS    LT11                    ;       SKIP
016644  010403                          MOV     R4,R3                   ; ELSE, USE ACTUAL BYTE COUNT
016646  013702  002160          LT11:   MOV     FFREE,R2                ; GET ADDRESS OF BUFFER
016652  162702  000214                  SUB     #<HC.BSZ+2>,R2          ;
016656                                  PUSH    <R3,R2>                 ; SAVE THE BYTE COUNT & BUFFER ADDRESS
016662  012022          LT1L2:  MOV     (R0)+,(R2)+             ; MOVE DATA TO BUFFER
016664  162703  000002                  SUB     #2.,R3                  ; DECREMENT WORDS REMAINING
016670  001374                          BNE     LT1L2                   ; LOOP IF MORE WORDS TO LOAD
016672                                  POP     <R2,R3>                 ; RESTORE BUFFER ADDRESS
016676  010037  002340                  MOV     R0,DMPTR                ; SAVE POINTER INTO DM PROGRAM
016702  012700  000031                  MOV     #OP.MWR,R0              ; GET DOWNLINE LOAD COMMAND
016706  004737  017056                  CALL    LOAD                    ; LOAD INTO CONTROLLER
```

```
     53 016712  001457                          BEQ     LOADER                  ; IF ERROR, GET OUT NOW
     54 016714  013700  002340                   MOV     DMPTR,R0                ; RESTORE POINTER INTO DM PROGRAM
     55 016720  006203                          ASR     R3                      ; CONVERT BYTES TO WORDS
     56 016722  060301                          ADD     R3,R1                   ; INCREASE OFFSET FOR NEXT BUFFER
     57 016724  006303                          ASL     R3                      ; CONVERT WORDS TO BYTES
     58 016726  160304                          SUB     R3,R4                   ; REDUCE REMAINING BYTE COUNT
     59 016730  001341                          BNE     LT1L1                   ; GET NEXT BUFFER
     60 016732  012701  000040                   MOV     #DMMAIN,R1              ; GET A BYTE COUNT OF HEADER ONLY
     61 016736  000402                          BR      LOADB                   ; NOW START
     63
     64 016740  017701  163234          LOADTX: MOV     @DMPROG,R1              ; GET SIZE OF PROGRAM
     65 016744  012700  000002          LOADB:  MOV     #OP.ESP,R0              ; BUILD EXECUTE SUPPLIED
     66 016750  004737  017176                  CALL    BLDCMD                  ;       PROGRAM COMMAND PACKET
     67 016754  013764  002200  000040          MOV     DMPROG,HC.CPK+P.UADR(R4); LOAD MAIN PROGRAM ADDRESS
     68 016762  010164  000034                  MOV     R1,HC.CPK+P.BCNT(R4)    ;       AND SIZE
     69 016766  013764  002200  000054          MOV     DMPROG,HC.CPK+P.OVRL(R4); LOAD OVERLAY ADDRESS
     70 016774  067764  163200  000054          ADD     @DMPROG,HC.CPK+P.OVRL(R4)
     78 017002  004737  017270                  CALL    SNDCMD                  ; SEND COMMAND TO CONTROLLER
     79 017006  004737  017400                  CALL    WAITMS                  ; WAIT FOR MESSAGE RESPONSE
     80 017012  001417                          BEQ     LOADER                  ; EXIT IF ERROR
     81 017014  032764  000037  000032          BIT     #ST.MSK,HC.MPK+P.STS(R4); CHECK FOR ERRORS
     82 017022  001007                          BNE     LOADE1                  ;
     83 017024  042765  000024  000012          BIC     #CT.CMD+CT.REQ,C.FLG(R5); CLEAR COMMAND OUTSTANDING FLAG
     84 017032  052765  000002  000012          BIS     #CT.RN,C.FLG(R5)        ; SET DM PROGRAM RUNNING FLAG
     85
     86 017040  000207                          RETURN                          ; SUCCESS RETURN
     87                                 ;
     88                                 ;       CONTROLLER FAILED TO DOWNLINE LOAD DM PROGRAM
     89                                 ;
     90
     91 017042                          LOADE1: ERRDF   33,,ERR033              ;
        017042  104455                          TRAP    C$ERDF
        017044  000041                          .WORD   33
        017046  000000                          .WORD   0
        017050  011474                          .WORD   ERR033
     92 017052  000264                  LOADER: SEZ                             ; SET Z TO INDICATE ERROR OCCURRED
     93 017054  000207                          RETURN                          ; ERROR RETURN
     94
```

```
                                ;++
                                ;       LOAD - ISSUE MAINTENANCE I/O COMMAND TO CONTROLLER. CHECK THAT I/O
                                ;       HAPPENS WITHOUT ERROR.
                                ;
                                ;       INPUTS:
                                ;               R0 - MAINTENANCE I/O COMMAND
                                ;               R1 - OFFSET INTO CONTROLLER MEMORY
                                ;               R2 - ADDRESS OF MESSAGE BUFFER
                                ;               R3 - SIZE OF BUFFER IN BYTES
                                ;               R5 - CONTROLLER TABLE ADDRESS
                                ;       OUTPUTS:
                                ;               Z CLEAR IF NO ERROR
                                ;               Z SET IF ERROR AND EPROR REPORTED
                                ;--
                                LOAD:   PUSH    <R0,R3,R4>
017056                                  CALL    BLDCMD                  ; BUILD COMMAND PACKET
017064  004737  017176                  MOV     R2,HC.CPK+P.UADR(R4)    ; STUFF IN BUFFER ADDRESS
017070  010264  000040                  MOV     R3,HC.CPK+P.BCNT(R4)    ; STUFF IN BYTE COUNT
017074  010364  000034                  MOV     R1,HC.CPK+P.RGOF(R4)    ; STUFF IN OFFSET
017100  010164  000060                  MOV     #1,HC.CPK+P.RGID(R4)    ; STUFF IN REGION ID 1
017104  012764  000001  000054          CALL    SNDCMD                  ; SEND COMMAND TO CONTROLLER
017112  004737  017270                  CALL    WAITMS                  ; WAIT FOR MESSAGE RESPONSE
017116  004737  017400                  BEQ     LOADX                   ; IF FAILED, EXIT
017122  001420                          BIT     #ST.MSK,HC.MPK+P.STS(R4); LOOK FOR ANY ERROR
017124  032764  000037  000032          BNE     LOADX1                  ;
017132  001010                          BIC     #CT.CMD,C.FLG(R5)       ; CLEAR COMMAND ISSUED
017134  042765  000004  000012          POP     <R4,R3,R0>              ; RESTORE REGISTERS
017142                                  CLZ                             ; CLEAR Z TO INDICATE NO ERROR
017150  000244                          RETURN                          ;
017152  000207
                                ;
                                ;       CONTROLLER FAILED TO DOWNLINE LOAD DM PROGRAM
                                ;
017154                          LOADX1: ERRDF   33,,ERR033              ; PRINT ERROR INFO FROM CONTROLLER
017154  104455                          TRAP    C$ERDF
017156  000041                          .WORD   33
017160  000000                          .WORD   0
017162  011474                          .WORD   ERR033
017164                          LOADX:  POP     <R4,R3,R0>              ; RESTORE STACK AFTER ERROR
017172  000264                          SEZ                             ; SET Z TO INDICATE ERROR OCCURRED
017174  000207                          RETURN
```

```
 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 017176                               BLDCMD: PUSH    <R1,R0>                 ; SAVE REGISTERS ON STACK
18 017202  016504  000014                       MOV     C.HCOM(R5),R4           ; GET ADDRESS OF HOST COMM AREA
19 017206  010400                               MOV     R4,R0                   ; COPY TO R0
20 017210  062700  000014                       ADD     #HC.CEV,R0              ; COMPUTE ADDRESS OF COMMAND ENVELOPE
21 017214  012720  000060                       MOV     #HC.PS2,(R0)+           ; LOAD PACKET LENGTH
22 017220  012701  001000                       MOV     #DUP,R1                 ; LOAD DUP CIRCUIT IDENTIFIER
23 017224  022716  000031                       CMP     #OP.MWR,(SP)            ; IF CODE IS MAINTENANCE WRITE
24 017230  001403                               BEQ     1$                      ;       USE DIAGNOSTIC CIRCUIT ID
25 017232  022716  000030                       CMP     #OP.MRD,(SP)            ; IF CODE IS NOT MAINTENANCE READ
26 017236  001002                               BNE     BLDC0                   ;       SKIP
27 017240  012701  177777               1$:     MOV     #DIAG,R1                ; ELSE, USE DIAGNOSTIC CIRCUIT ID
28 017244  010120                       BLDC0:  MOV     R1,(R0)+                ; PUT IDENTIFIER INTO PACKET
29 017246  012701  000030                       MOV     #<HC.PSZ>/2,R1          ; GET WORDS TO CLEAR
30 017252  005020                       BLDC1:  CLR     (R0)+                   ; CLEAR PACKET
31 017254  005301                               DEC     R1                      ; ANY MORE
32 017256  001375                               BNE     BLDC1                   ;       WORDS TO CLEAR?
33 017260                                       POP     HC.CPK+P.OPCD(R4)       ; PUT OPCODE IN PACKET
34 017264                                       POP     R1                      ; RESTORE R1
35 017266  000207                               RETURN
36
```

```
 1
 2                                      ;**
 3                                      ;       SNDCMD - SEND A COMMAND TO THE CONTROLLER. CLEAR THE RESPONSE
 4                                      ;       PACKET. MARK BOTH PACKETS AVAILABLE TO THE CONTROLLER. SET COMMAND
 5                                      ;       ISSUED BIT IN CONTROLLER TABLE AND INITIALIZE TIMEOUT COUNTER.
 6                                      ;
 7                                      ;       INPUTS:
 8                                      ;               R5 - CONTROLLER TABLE ADDRESS
 9                                      ;       OUTPUTS:
10                                      ;               R4 - ADDRESS OF HOST COMM AREA
11                                      ;--
12
13 017270  016504  000014          SNDCMD: MOV     C.HCOM(R5),R4           ; LOAD R4 WITH HOST COMM AREA ADDRESS
14 017274  005265  000032                  INC     C.REF(R5)               ; INCREMENT CMD REFERENCE NUMBER
15 017300  016564  000032  000020          MOV     C.REF(R5),HC.CPK+P.CRF(R4); PUT IN PACKET
16 017306  012764  140000  000006          MOV     #RG.OWN+RG.FLG,HC.MCT(R4) ; MARK MESSAGE PACKET AVAILABLE
17 017314  012764  100000  000012          MOV     #RG.OWN,HC.CCT(R4)      ; MARK COMMAND TO CONTROLLER
18 017322  005775  000000                  TST     @(R5)                   ; TELL CONTROLLER COMMAND IS THERE
19 017326  052765  000004  000012          BIS     #CT.CMD,C.FLG(R5)       ; MARK COMMAND ISSUED
20 017334  000207                          RETURN
21
```

```
                                        ;++
                                        ;       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
                                        ;--

017336                          CLRBUF: PUSH    <R0,R1>                 ; SAVE REGISTERS ON STACK
017342  060400                          ADD     R4,R0                   ; CREATE BUFFER ADDRESS
017344  010064  000040                  MOV     R0,MC.CPK+P.UADR(R4)    ; PUT BUFFER ADDRESS IN COMMAND PACKET
017350  012764  000106  000034          MOV     #MC.BSZ,MC.CPK+P.BCNT(R4);PUT SIZE OF BUFFER IN COMMAND PACKET
017356  010004                          MOV     R0,R4                   ; PUT BUFFER ADDRESS IN R4
017360  012701  000043                  MOV     #<MC.BSZ>/2,R1          ; GET SIZE OF BUFFER IN WORDS
017364  005020                  CLR8FL: CLR     (R0)+                   ; CLEAR ALL THE WORDS
017366  005301                          DEC     R1                      ; ANY MORE .
017370  001375                          BNE     CLR8FL                  ;       WORDS TO CLEAR?
017372                                  POP     <R1,R0>                 ;
017376  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
                                        ;--

017400                          WAITMS: PUSH    <R0,R1>                 ; SAVE REGISTERS ON STACK
017404  012700  000036                  MOV     #30.,R0                 ; SET TIME OUT VALUE OF 30 SECONDS
017410  010501                          MOV     R5,R1                   ; POINT TO TIME OUT COUNTER
017412  062701  000026                  ADD     #C.TO,R1                ; POINTER TO TIMER FIELD
017416  004737  017614                  CALL    SETTO                   ; START TIMER
017422  011500                  1$:     MOV     (R5),R0                 ; GET ADDRESS OF IP REGISTER
017424  032765  000010  000012          BIT     #CT.MSG,C.FLG(R5)       ; LOOK IF INTERRUPT OCCURRED
017432  001025                          BNE     3$                      ; BRANCH IF SO
017434  016001  000002                  MOV     2(R0),R1                ; LOOK AT SA REGISTER
017440  001031                          BNE     4$                      ; BRANCH IF ERROR CODE PRESENT
017442                                  BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
017442  104422                          TRAP    C$BRK
017444  005737  002256                  TST     KW.CSR                  ; SEE IF A CLOCK ON SYSTEM
017450  001764                          BEQ     1$                      ; IF NOT, DON'T TIMEOUT
017452  023765  002270  000030          CMP     KW.EL+2,C.TOH(R5)       ; CHECK HIGH WORD OF INTERVAL
017460  101005                          BHI     2$                      ; IF GREATER, TIMED OUT
017462  001357                          BNE     1$                      ; IF NOT EQUAL, NO TIMEOUT OCCURRED
017464  023765  002266  000026          CMP     KW.EL,C.TO(R5)          ; CHECK LOW WORD OF INTERVAL
017472  103753                          BLO     1$                      ; IF LOWER, NO TIMEOUT OCCURRED
017474                          2$:     ERRDF   36.,ERR036              ; PRINT TIMEOUT ERROR
017474  104455                          TRAP    C$ERDF
017476  000044                          .WORD   36
017500  000000                          .WORD   0
017502  011520                          .WORD   ERR036
017504  000413                          BR      5$                      ; ERROR EXIT

017506  042765  000010  000012  3$:     BIC     #CT.MSG,C.FLG(R5)       ; CLEAR MESSAGE RECEIVED FLAG
017514                                  POP     <R1,R0>                 ; SAVE REGISTERS ON STACK
017520  000244                          CLZ                             ; GIVE NO ERROR RETURN
017522  000207                          RETURN                          ;
017524                          4$:
017524                                  ERRDF   30.,ERR030              ; CONTROLLER DETECTED ERROR
017524  104455                          TRAP    C$ERDF
017526  000036                          .WORD   30
017530  000000                          .WORD   0
017532  011430                          .WORD   ERR030
017534                          5$:     POP     <R1,R0>                 ; RESTORE REGISTERS
017540  000264                          SEZ                             ; Z SET OR ERROR RETURN
017542  000207                          RETURN                          ;
```

```
                                        ;++
                                        ;       NXMI - NON-EXISTANT MEMORY SERVICE ROUTINE
                                        ;
                                        ;       INPUTS:
                                        ;               NXMAD SET TO ZERO
                                        ;       OUTPUTS:
                                        ;               NXMAD SET TO ONES IF NON-EXISTANT TRAP OCCURED
                                        ;--

017544                                  BGNSRV  NXMI
017544                                  NXMI::
017544  012737  177777  002276                  MOV     #-1,NXMAD
017552                                  ENDSRV
017552                                  L10033:
017552  000002                                  RTI

                                        ;++
                                        ;       CNTSRV - CONTROLLER INTERRUPT SERVICE ROUTINE. MARKS CONTROLLER TABLE
                                        ;       THAT AN INTERRUPT HAS BEEN RECEIVED.
                                        ;
                                        ;       THIS ROUTINE IS CALLED BY A [JSR R0,CNTSRV] INSTRUCTION FROM WITHIN
                                        ;       THE CONTROLLER TABLE. THE PC STORED IN R0 IS THE ADDRESS OF THE C.FLG
                                        ;       WORD IN THE CONTROLLER TABLE. THE STACK CONTAINS THE SAVED CONTENTS
                                        ;       OF R0 FOLLOWED BY THE INTERRUPTED PC AND PS.
                                        ;
                                        ;       INPUTS:
                                        ;               R0 - ADDRESS OF C.FLG WORD IN CONTROLLER TABLE
                                        ;               STACK - SAVED CONTENTS OF R0
                                        ;       OUTPUTS:
                                        ;               CT.MSG SET IN C.FLG WORD OF CONTROLLER TABLE
                                        ;               R0 - RESTORED FROM STACK
                                        ;--

017554                                  BGNSRV  CNTSRV
017554                                  CNTSRV::
017554  052710  000010                          BIS     #CT.MSG,(R0)            ; SET CT.MSG
017560                                          POP     R0                      ; RESTORE R0
017562                                  ENDSRV
017562                                  L10034:
017562  000002                                  RTI

                                        ;++
                                        ;       INTSRV - CONTROLLER INTERRUPT SERVER
                                        ;--

017564                                  BGNSRV  INTSRV
017564                                  INTSRV::
017564  005237  002214                          INC     INTRCV                  ; FLAG INTERRUPT AS RECEIVED
017570                                  ENDSRV
017570                                  L10035:
017570  000002                                  RTI

                                        ;++
                                        ;       KW11I - CLOCK INTERRUPT SERVICE ROUTINE
                                        ;--
017572                                  BGNSRV  KW11I
```

```
                017572                          KW11I::
     51 017572  062737  000001  002266                  ADD     #1,KW.EL                ; COUNT THE INTERRUPT
     52 017600  005537  002270                          ADC     KW.EL+2                 ; PUT CARRY IN HIGH WORD
     53 017604  012777  000105  162444                  MOV     #KW.OUT,@KW.CSR         ; RESTART THE CLOCK
     54 017612                          ENDSRV
                017612                          L10036:
                017612  000002                          RTI
     55
```

```
 1
 2                                      ;++
 3                                      ;       SETTO   SET TIMEOUT COUNTER TO A GIVEN NUMBER OF SECONDS FROM CURRENT
 4                                      ;               TIME.
 5                                      ;
 6                                      ;       INPUTS:
 7                                      ;               R0 - NUMBER OF SECONDS FOR TIMEOUT
 8                                      ;               R1 - ADDRESS WHERE TWO WORD TIME TO BE PUT
 9                                      ;       OUTPUTS:
10                                      ;               R0 - CONTENTS DESTROYED
11                                      ;               R1 - INCREMENTED BY 2
12                                      ;--
13
14 017614                               SETTO:  PUSH    <R2,R3>
15 017620  005002                               CLR     R2                      ; CLEAR PRODUCT
16 017622  013703  002264                       MOV     KW.HZ,R3                ; GET MULTIPLICAND
17
18 017626  006200                       SETOO:  ASR     R0                      ; SHIFT MULTIPLIER TO RIGHT
19 017630  103001                               BCC     SET01                   ; IF A ONE BIT SHIFTED OUT
20 017632  060302                               ADD     R3,R2                   ;   ADD MULTIPLICAND TO PRODUCT
21 017634  006303                       SET01:  ASL     R3                      ; DOUBLE THE MULTIPLICAND
22 017636  005700                               TST     R0                      ;
23 017640  001372                               BNE     SETOO                   ; CONTINUE UNTIL MULTIPLIER IS ZERO
24
25                                      ;
26                                      ;       GET CURRENT TIME
27                                      ;
28
29 017642  013700  002266               SET02:  MOV     KW.EL,R0                ; GET TIME (LOW WORD)
30 017646  013703  002270                       MOV     KW.EL+2,R3              ; GET TIME (HIGH WORD)
31 017652  020037  002266                       CMP     R0,KW.EL                ; IF CHANGED DURING RETRIEVAL
32 017656  001371                               BNE     SET02                   ;   GET IT AGAIN
33
34                                      ;
35                                      ;       ADD TIME TIL TIMEOUT
36                                      ;
37
38 017660  060200                               ADD     R2,R0                   ; ADD TIMEOUT TO CURRENT TIME
39 017662  005503                               ADC     R3                      ; INCREMENT HIGH WORD IF CARRY
40
41                                      ;
42                                      ;       PUT RESULT IN STORAGE
43                                      ;
44
45 017664  010021                               MOV     R0,(R1)+                ; SAVE LOW WORD OF TIMEOUT
46 017666  010311                               MOV     R3,(R1)                 ; SAVE HIGH WORD OF TIMEOUT
47
48 017670                                       POP     <R3,R2>
49 017674  000207                               RETURN
50
```

```
                                ;**
                                ;       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
                                ;
017676                          CNTINT: PUSH    <R3>                    ; SAVE R3 ON STACK
017700  010400                          MOV     R4,R0                   ; GET MESSAGE LENGTH
017702  000300                          SWAB    R0                      ; RIGHT JUSTIFY MESSAGE LENGTH
017704  042700  177770                 BIC     #177770,R0              ; REMOVE EXTRA INFO
017710  004737  020724                 CALL    CLOG                    ; COMPUTE POWER OF 2
017714  010102                         MOV     R1,R2                   ; SAVE MESSAGE LENGTH+2  IN R2
017716  010400                         MOV     R4,R0                   ; GET COMMAND LENGTH
017720  000300                         SWAB    R0                      ; RIGHT
017722  006000                         ROR     R0                      ;     JUSTIFY
017724  006000                         ROR     R0                      ;           COMMAND
017726  006000                         ROR     R0                      ;                 LENGTH
017730  042700  177770                 BIC     #177770,R0              ; REMOVE EXTRA INFO
017734  004737  020724                 CALL    CLOG                    ; COMPUTE POWER OF 2
017740  060201                         ADD     R2,R1                   ; ADD THE TWO RESULTS
017742  006301                         ASL     R1                      ; MULTIPLY BY 2 WORDS PER RING
017744  062701  000002                 ADD     #<HC.ISZ>/2,R1          ; ADD SPACE FOR INTERRUPT INDICATORS
017750  020137  002162                 CMP     R1,FSIZE                ; COMPARE WITH SIZE OF FREE MEMORY
017754  101402                         BLOS    1$                      ; THERE IS PLENTY OF MEMORY
017756  000137  012120                 JMP     FMERR                   ; FATAL ERROR IF NOT ENOUGH MEMORY
017762                          1$:
                                ;
                                ;       FILL HOST COMMUNICATION AREA WITH ALL ONES
                                ;
017762  013702  002160                 MOV     FFREE,R2                ; GET FIRST ADDRESS OF RING BUFFER
017766  010103                         MOV     R1,R3                   ; GET SIZE OF RING BUFFER
017770  012722  177777         2$:     MOV     #-1,(R2)+               ; WRITE ONES TO BUFFER
017774  005303                         DEC     R3                      ; COUNT THE WORDS IN BUFFER
017776  003374                         BGT     2$                      ; LOOP UNTIL ENTIRE BUFFER WRITTEN
```

```
                                ;
                                ;       DO THE INITIALIZATION
                                ;
020000  004737  020272          CALL    CNTIST                  ; DO FIRST THREE STEPS
020004  103527                  BCS     9$                      ; EXIT IF MICROCODE REPORTED FAILURE
020006  012364  000002          MOV     (R3)+,2(R4)             ; WRITE STEP 3 RESPONSE TO SA REGISTER
020012  012700  000310          MOV     #200.,R0                ; GET TRY COUNTER
020016  016^02  000002  3$:     MOV     2(R4),R2                ; LOOK AT SA REGISTER
020022  001407                  BEQ     5$                      ; EXIT LOOP IF ZERO
020024  005300                  DEC     R0                      ;
020026  001373                  BNE     3$                      ; KEEP LOOPING IF NOT ZERO
020030                          ERRDF   24.,ERR024              ;
020030  104455                  TRAP    C$ERDF
020032  000030                  .WORD   24
020034  000000                  .WORD   0
020036  011312                  .WORD   ERR024
020040  000511                  BR      9$                      ;
020042  005064  000002  5$:     CLR     2(R4)                   ; WRITE 0 TO SA REGISTER (PURGE)
020046  005714                  TST     (R4)                    ; READ FROM IP REGISTER (POLL)
020050  004737  020566          CALL    CNTRSP                  ; WAIT FOR STEP OR ERROR BIT
020054  103503                  BCS     9$                      ; EXIT IF MICROCODE REPORTED FAILURE
020056  010237  002326          MOV     R2,SSTEP4               ; SAVE STEP 4 RESPONSE VALUE.
                                ;
                                ;       CHECK HOST COMMUNICATION AREA FOR ALL ZEROS
                                ;
020062  013702  002160          MOV     FFREE,R2                ; GET FIRST ADDRESS OF RING BUFFER
020066  010103                  MOV     R1,R3                   ; GET SIZE OF RING BUFFER
020070  005722          6$:     TST     (R2)+                   ; CHECK WORD IN BUFFER
020072  001003                  BNE     7$                      ; GO TO ERROR REPORTER IF NOT ZERO
020074  005303                  DEC     R3                      ; COUNT THE WORDS IN BUFFER
020076  003374                  BGT     6$                      ; LOOP UNTIL ALL WORDS CHECKED
020100  000405                  BR      8$                      ; START CONTROLLER AND EXIT

020102                  7$:     ERRDF   23.,ERR023              ; REPORT BUFFER NOT CLEARED
020102  104455                  TRAP    C$ERDF
020104  000027                  .WORD   23
020106  000000                  .WORD   0
020110  011232                  .WORD   ERR023
020112  000464                  BR      9$                      ; ERROR EXIT
                                ;
                                ;       SEND GO BIT TO SA REGISTER TO END INITIALIZATION
                                ;
020114                  8$:
020114  012700  000001          MOV     #SA.GO,R0               ; SET THE GO BIT
020120  010064  000002          MOV     R0,2(R4)                ; WRITE TO SA REGISTER
                                ;
                                ;       NOW CHECK IF THE CONTROLLER TYPE IS VALID AND MICROCODE VERSION
                                ;       IS CURRENT
                                ;
020124  013703  002326  23$:     MOV     SSTEP4,R3               ; GET SAVED CONTROLLER STEP 4 RESPONSE
020130  010301                  MOV     R3,R1                   ; R3 HAS STEP 4 INFO
020132  042701  177760          BIC     #^C<SA.MCV>,R1          ; R1 = MICRO CODE LEVEL
020136  006003                  ROR     R3                      ; RIGHT
020140  006003                  ROR     R3                      ;       JUSTIFY
020142  006003                  ROR     R3                      ;               MODEL
020144  006003                  ROR     R3                      ;                       NUMBER
020146  042703  177760          BIC     #^C<SA.CNT/16.>,R3      ; R3 = CONTROLLER MODEL NUMBER
```

```
                                        CMP     #6.,R3                  ; CHECK IF UDA50A
                                        BEQ     24$                     ; IF SO, BRANCH
                                        CMP     #13.,R3                 ; CHECK IF KDA50-Q
                                        BEQ     25$                     ; IF SO BRANCH
                                        BIS     #BIT15,C.UNIT(R5)       ;       ELSE MARK AS DO NOT EXECUTE
                                        ERRDF   14.,ERR014              ;       REPORT ERROR
                                        TRAP    C$ERDF
                                        .WORD   14
                                        .WORD   0
                                        .WORD   ERR014
                                        BR      9$                      ;       EXIT
                                24$:
                                        CLR     SSTEP4                  ; CLEARED FOR TEST 1, SUBTEST 4
                                        CMP     R1,#3.                  ; UDA50A MICROCODE VERSION UP TO DATE?
                                        BGE     26$                     ; IF SO, BRANCH
                                        ERRSF   14.,ERR014              ;       ELSE, REPORT ERROR
                                        TRAP    C$ERSF
                                        .WORD   14
                                        .WORD   0
                                        .WORD   ERR014
                                        BR      26$                     ;
                                25$:
                                        MOV     #4.,SSTEP4              ; DEFAULT BR LVL, TST 1,SUBTST 4
                                        CMP     R1,#0.                  ; KDA50-Q MICROCODE VERSION UP TO DATE?
                                        BGE     26$                     ; IF SO, BRANCH
                                        ERRSF   14.,ERR014              ;       ELSE, REPORT ERROR
                                        TRAP    C$ERSF
                                        .WORD   14
                                        .WORD   0
                                        .WORD   ERR014
                                26$:    POP     <R3>                    ; RESTORE R3 FROM STACK
                                        CLZ                             ; CLEAR Z TO INDICATE NO ERROR
                                        RETURN                          ;

                                ;
                                ;       ERROR RETURN
                                ;

                                9$:     POP     <R3>                    ; RESTORE R3 FROM STACK
                                        SEZ                             ; SET Z TO INDICATE ERROR OCCURRED
                                        RETURN                          ;
```

```
 1
 2                                      ;++
 3                                      ;       CNTIST
 4                                      ;
 5                                      ;       START THE INITIALIZATION PROCESS ON THE SELECTED CONTROLLER.
 6                                      ;       STOP BEFORE WRITING THE THIRD WORD SO CONTROLLER DOES NOT
 7                                      ;       ATTEMPT ANY BUS TRANSFERS.
 8                                      ;
 9                                      ;       INPUTS:
10                                      ;               R5 - ADDRESS OF CONTROLLER TABLE
11                                      ;               R4 - LEN, INTI AND VECTOR FIELDS TO SEND TO CONTROLLER
12                                      ;
13                                      ;       LOAD TABLE OF DATA TO SEND TO SA REGISTER
14                                      ;--
15
16 020272                              CNTIST:
17 020272                                       BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
   020272  104422                               TRAP    C$BRK
18 020274                                       PUSH    R1                      ;
19 020276  052704  100000                       BIS     #SA.STP,R4              ; SET STEP BIT IN DATA WORD
20 020302  010437  002314                       MOV     R4,SND.S1               ; LOAD SEND DATA FOR STEP 1 OF INIT
21 020306  013737  002160  002320               MOV     FFREE,SND.S2            ; GET MEMORY ADDRESS AND
22 020314  062737  000004  002320               ADD     #MC.MSG,SND.S2          ; LOAD SEND DATA FOR STEP 2 OF INIT
23 020322  012737  100000  002324               MOV     #SA.TST,SND.S3          ; LOAD SEND DATA FOR STEP 3 OF INIT
24                                      ;
25                                      ;       VERIFY THE ADDRESS OF THE SA AND IP REGISTERS ARE VALID AND
26                                      ;       START THE INITIALIZATION BY WRITING ZEROS TO IP REGISTER
27                                      ;
28 020330  016504  000000                       MOV     C.UADR(R5),R4           ; GET ADDRESS OF IP REGISTER
29 020334  005037  002276                       CLR     NXMAD                   ; CLEAR MEMORY ERROR FLAG
30 020340                                       SETVEC  #ERRVEC,#NXMI,#PRI07    ; SETUP TIMEOUT ERROR VECTOR
   020340  012746  000340                       MOV     #PRI07,-(SP)
   020344  012746  017544                       MOV     #NXMI,-(SP)
   020350  012746  000004                       MOV     #ERRVEC,-(SP)
   020354  012746  000003                       MOV     #3,-(SP)
   020360  104437                               TRAP    C$SVEC
   020362  062706  000010                       ADD     #10,SP
31 020366  005764  000002                       TST     2(R4)                   ; ACCESS SA REGISTER
32 020372  005014                               CLR     (R4)                    ; WRITE TO IP
33 020374                                       CLRVEC  #ERRVEC                 ; RETURN TIMEOUT ERROR VECTOR
   020374  012700  000004                       MOV     #ERRVEC,R0
   020400  104436                               TRAP    C$CVEC
34 020402  005737  002276                       TST     NXMAD                   ; SEE IF A MEMORY ERROR OCCURRED
35 020406  001406                               BEQ     1$                      ; IF NO ERROR, CONTINUE
36 020410                                       ERRDF   38.,ERR038              ; PRINT ERROR MESSAGE
   020410  104455                               TRAP    C$ERDF
   020412  000046                               .WORD   38
   020414  000000                               .WORD   0
   020416  011532                               .WORD   ERR038
37 020420  000261                               SEC                             ; CARRY SET TO INDICATE ERROR
38 020422  000424                               BR      4$                      ; EXIT
39
40                                      ;
41                                      ;       SET UP LOOP PARAMETERS TO EXECUTE THE FOUR STEPS OF INITIALIZATION
42                                      ;
43
44 020424  012737  004000  002330  1$:  MOV     #SA.S1,CNTRSD                   ; STORE RESPONSE MASK
```

```
                                                MOV     #INITBL,R3              ; GET INDEX TO SEND/REPOND INIT TABLE
                                ;
                                ;       WAIT FOR AND CHECK RESPONSE DATA
                                ;
                        2$:     CALL    CNTRSP                  ; WAIT FOR STEP OR ERROR BITS
                                BCS     4$                      ; EXIT IF ERROR
                                CALL    @(R3)+                  ; CALL RESPONSE CHECKER FOR STEP
                                BCS     4$                      ; GET OUT IF ERROR
                                ASL     CNTRSD                  ; SHIFT TO NEXT STEP BIT
                                BIT     #SA.S4,CNTRSD           ; CHECK IF NOW AT STEP 4
                                BNE     3$                      ; GET OUT IF SO
                                MOV     (R3)+,2(R4)             ; WRITE DATA TO SA REGISTER
                                BR      2$                      ; STAY IN LOOP

                        3$:     CLC                             ; CLEAR CARRY FOR NO ERROR INDICATION
                        4$:     POP     R1                      ; RESTORE R1
                                RETURN                          ;
                                ;
                                ;       RESPONSE CHECK FOR FIRST WORD (STEP 1) FROM SA REGISTER
                                ;       CHECK FOR PROPER CONTROLLER TYPE
                                ;
                        RSP.S1: MOV     #SA.S1+SA.DI,R1         ; SET STEP ONE BIT
                                BIC     #SA.QB+SA.MP+SA.SM,R2   ; CLEAR Q22 & SM BIT FOR KDA50-Q
                                BR      RSP.CK                  ; NOW DO A RESPONSE CHECK
                                ;
                                ;       RESPONSE CHECK FOR SECOND WORD (STEP 2) FROM SA REGISTER
                                ;       CHECK FOR ECHO OF INTERRUPT ENABLE FLAG AND INTERRUPT VECTOR
                                ;
                        RSP.S2: MOV     SND.S1,R1               ; GET WORD SENT TO SA REGISTER
                                SWAB    R1                      ; GET HIGH 8 BITS
                                BIC     #177400,R1              ;
                                BIS     #SA.S2,R1               ; SET STEP 2 BIT
                                BR      RSP.CK                  ; NOW DO A RESPONSE CHECK
                                ;
                                ;       RESPONSE CHECK FOR THIRD WORD (STEP 3) FROM SA REGISTER
                                ;       CHECK FOR ECHO OF MESSAGE AND COMMAND RING LENGTHS
                                ;
                        RSP.S3: MOV     SND.S1,R1               ; GET WORD SENT TO SA REGISTER
                                BIC     #177400,R1              ; JUST LOW 8 BITS
                                BIS     #SA.S3,R1               ; SET STEP 3 BIT
                                ;
                                ;       RESPONSE CHECK, COMPARE EXPECTED DATA IN R1 WITH ACTUAL DATA IN R2
                                ;
                        RSP.CK: CMP     R1,R2                   ; COMPARE THE DATA
                                BEQ     1$                      ; EXIT IF COMPARED CORRECTLY
                                ERRDF   25,,ERR025              ; ERROR - WRONG DATA IN SA REGISTER
                                TRAP    C$ERDF
                                .WORD   25
                                .WORD   0
                                .WORD   ERR025
                                SEC                             ; SET CARRY TO INDICATE ERROR
                        1$:     RETURN                          ;
```

```
                                        ;**
                                        ;       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
                                        ;               P4 - 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
                                        ;--
020566                          CNTRSP: PUSH    R1
020570  052737  100000  002330          BIS     #SA.ERR,CNTRSD          ; SET ERROR BIT IN MASK WORD
020576  012700  000012                  MOV     #10.,R0                 ; SET UP FOR 10 SECOND TIMEOUT
020602  010501                          MOV     R5,R1                   ; POINT TO COUNTER IN CONTROLLER TABLE
020604  062701  000026                  ADD     #C.TO,R1                ;
020610  004737  017614                  CALL    SETTO                   ;
020614                                  POP     R1                      ;
020616  033764  002330  000002  1$:     BIT     CNTRSD,2(R4)            ; LOOK AT ERROR AND STEP BIT
020624  001024                          BNE     3$                      ; BRANCH IF EITHER SET
020626                                  BREAK                           ; >>>>>>>>BREAK BACK TO MONITOR<<<<<<<
020626  104422                          TRAP    C$BRK
020630  005737  002256                  TST     KW.CSR                  ; SEE IF CLOCK ON SYSTEM
020634  001770                          BEQ     1$                      ;
020636  023765  002270  000030          CMP     KW.EL+2,C.TOH(R5)       ; CHECK IF TIME OUT OCCURRED
020644  101005                          BHI     2$                      ;
020646  001353                          BNE     1$                      ;
020650  023765  002266  000026          CMP     KW.EL,C.TO(R5)          ;
020656  103757                          BLO     1$                      ;
020660  016402  000002          2$:     MOV     2(R4),R2                ; GET REGISTER CONTENTS
020664                                  ERRDF   22.,ERR022              ; REPORT TIME OUT ERROR
020664  104455                          TRAP    C$ERDF
020666  000026                          .WORD   22
020670  000000                          .WORD   0
020672  011204                          .WORD   ERR022
020674  000407                          BR      4$                      ;
                                        ;
                                        ;       CHECK IF ERROR BIT SET
                                        ;
020676  016402  000002          3$:     MOV     2(R4),R2                ; GET REGISTER CONTENTS
020702  100006                          BPL     5$                      ; EXIT IF ERROR NOT SET
020704                                  ERRDF   21.,ERR021              ; REPORT ERROR INFO
020704  104455                          TRAP    C$ERDF
020706  000025                          .WORD   21
020710  000000                          .WORD   0
020712  011134                          .WORD   ERR021
020714  000261                  4$:     SEC                             ;
020716  000207                          RETURN                          ;
```

```
                                        ;
                                        ;       NORMAL EXIT
                                        ;
020720  000241                  5$:     CLC                             ; CLEAR CARRY AS NO ERROR INDICATION
020722  000207                          RETURN                          ;
```

```
 1
 2                                      ;**
 3                                      ;       CLOG - COMPUTE SELECTED POWER OF 2.
 4                                      ;
 5                                      ;       INPUTS:
 6                                      ;               R0 - SELECTED POWER OF 2
 7                                      ;       OUTPUTS:
 8                                      ;               R1 - VALUE OF 2 RAISED TO POWER OF INPUT NUMBER
 9
10 020724                              CLOG:   PUSH    R0                      ; SAVE INPUT VALUE
11 020726  005001                              CLR     R1                      ; SET UP ZERO START VALUE
12 020730  000261                              SEC                             ; CARRY READY TO SHIFT IN
13 020732  006101                      1$:     ROL     R1                      ; SHIFT 1 TO LEFT
14 020734  005300                              DEC     R0                      ;    UNTIL R0
15 020736  100375                              BPL     1$                      ;       GOES NEGATIVE
16 020740                                      POP     R0                      ; RESTORE INPUT VALUE
17 020742  000207                              RETURN                          ;
```

```
 1                                      ;RESET
 2                                      ;
 3                                      ; RESET ALL CONTROLLERS IN THE CONTROLLER TABLES
 4                                      ;
 5                                      ; INPUTS:
 6                                      ;       IPADRS - CONTAINS ALL IP ADDRESSES
 7                                      ; OUTPUTS:
 8                                      ;       NONE
 9                                      ;
10
11
12 020744  005037  002276          RESET:  CLR     NXMAD                   ; CLEAR NON-EXISTANT MEMORY ADDRESS
13 020750                                  PUSH    <R3,R4>                 ; SAVE R3 AND R4 ON STACK
14 020754                                  SETVEC  #ERRVEC,#NXMI,#PRI07    ; SETUP TIMEOUT ERROR VECTOR
   020754  012746  000340                  MOV     #PRI07,-(SP)
   020760  012746  017544                  MOV     #NXMI,-(SP)
   020764  012746  000004                  MOV     #ERRVEC,-(SP)
   020770  012746  000003                  MOV     #3,-(SP)
   020774  104437                         TRAP    C$SVEC
   020776  062706  000010                  ADD     #10,SP
15 021002  012703  000004                  MOV     #4.,R3                  ; STORE MAXIMUM # OF CONTROLLERS IN R3
16 021006  012704  002246                  MOV     #IPADRS,R4              ; STORE IP ADDRESS IN R4
17 021012  005714                  1$:     TST     (R4)                    ; IS THERE AN ENTRY?
18 021014  001403                          BEQ     2$                      ; IF NOT, DONE
19 021016  005034                          CLR     @(R4)+                  ; INIT CONTROLLER
20 021020  005303                          DEC     R3                      ; MAKE SURE WE DO NOT EXTEND OVER AREA
21 021022  001373                          BNE     1$                      ; IF NOT DONE, BRANCH
22 021024  005737  002256          2$:     TST     KW.CSR                  ; SEE IF CLOCK PRESENT,
23 021030  001403                          BEQ     3$                      ; BRANCH IF NOT, ELSE
24 021032  012777  000105  161216         MOV     #KW.OUT,@KW.CSR         ; START THE CLOCK.
25 021040                          3$:     POP     <R4,R3>                 ; RESTORE R3,R4 FROM STACK
26 021044  000207                          RETURN                          ;
27
```

```
                                ;RNTIME
                                ;
                                ;PRINT RUNTIME
                                ;
                                ;INPUTS:
                                ;       KW.EL - CONTAINS ELAPSED TIME
                                ;       KW.HZ - HERTZ OF CLOCK
                                ;OUTPUTS:
                                ;       IF CLOCK ON SYSTEM:
                                ;          "  RUN TIME HH:MM:SS " PRINTED
                                ;       IF NO CLOCK: ONE SPACE IS PRINTED
                                ;
021046  005737  002256          RNTIME: TST     KW.CSR                  ;CHECK IF A CLOCK PRESENT
021052  001465                          BEQ     RNTIMX                  ;BRANCH IF NOT
021054                                  PUSH    <R0,R3,R4,R5>
021064  013703  002266                  MOV     KW.EL,R3                        ;GET ELAPSED TIME
021070  013704  002270                  MOV     KW.EL+2,R4
021074  013700  002264                  MOV     KW.HZ,R0                        ;GET SPEED OF CLOCK
021100  004737  016014                  CALL    DIVIDE                  ;COMPUTE SECONDS OF ELAPSED TIME
021104  012700  000074                  MOV     #60.,R0                 ;NOW DIVIDE BY 60
021110  004737  016014                  CALL    DIVIDE                  ; TO COMPUTE MINUTES
021114                                  PUSH    R5                              ;SAVE REMAINDER AS SECONDS
021116  004737  016014                  CALL    DIVIDE                  ;DIVIDE BY 60 AGAIN
021122                                  PNT     RNTIM,R3                        ;PRINT HOURS
021134  020527  000011                  CMP     R5,#9.                  ;IF MINUTES 9 OR LESS
021140  003004                          BGT 1$
021142                                  PRINT   #'0                     ;PRINT A LEADING ZERO
021152                          1$:     PNT     RNTIM1,R5                       ;NOW PRINT MINUTES
021164                                  POP     R5                              ;GET SECONDS
021166  020527  000011                  CMP     R5,#9.                  ;IF 9 OR LESS
021172  003004                          BGT 2$
021174                                  PRINT   #'0                     ;PRINT A LEADING ZERO
021204                          2$:     PNT     RNTIM2,R5                       ;NOW PRINT SECONDS
021216                                  POP     <R5,R4,R3,R0>           ;HOURS IN R3
021226                          RNTIMX: PRINT   <#' >                   ;PRINT A SPACE
021236  000207                          RETURN
```

```
 1
 2                                      ; *
 3                                      ;       WCHNG
 4                                      ;
 5                                      ;       WAIT UNTIL LOOPED DATA APPEARS IN SA REGISTER
 6                                      ;--
 7
 8 021240  012700  000012          WCHNG:  MOV     #10.,R0                 ; SET TIMEOUT FOR 10 SECONDS
 9 021244  010501                          MOV     R5,R1                   ; POINT TO CONTROLLER TABLE
10 021246  062701  000026                  ADD     #C.TO,R1                ;
11 021252  004737  017614                  CALL    SETTO                   ;
12 021256  026437  000002  002332  1$:     CMP     2(R4),WCHNGD            ; IS LOOPED DATA IN SA REGISTER?
13 021264  001422                          BEQ     2$                      ;
14 021266                                  BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
   021266  104422                          TRAP    C$BRK
15 021270  005737  002256                  TST     KW.CSR                  ; SEE IF CLOCK ON SYSTEM
16 021274  001770                          BEQ     1$                      ;
17 021276  023765  002270  000030          CMP     KW.EL+2,C.TOH(R5)       ;CHECK IF TIME OUT OCCURRED
18 021304  101005                          BHI     3$                      ;
19 021306  001363                          BNE     1$                      ;
20 021310  023765  002266  000026          CMP     KW.EL,C.TO(R5)          ;
21 021316  103757                          BLO     1$                      ;
22 021320                          3$:     ERRDF   27,,ERR027              ; REPORT ERROR
   021320  104455                          TRAP    C$ERDF
   021322  000033                          .WORD   27
   021324  000000                          .WORD   0
   021326  011364                          .WORD   ERR027
23 021330  000244                          CLZ                             ; FLAG AS ERROR
24 021332  000207                  2$:     RETURN                          ; RETURN TO CALLING PROGRAM
26
```

```
                                .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".
                                ;--

                                        BGNRPT
021334                          L$RPT::

021334                                  PUSH    <R0,R1,R2,R3,R4,R5>
021350                                  PNTS    RPTMSG,TNUM             ; PRINT TEST NUMBER
021364  004737  021046                  CALL    RNTIME                  ; GET RUNTIME PARAMETERS
021370                                  PRINT   #CR                     ; END THE LINE
021400  012701  002272                  MOV     #STIME,R1               ; GET REPORT TIMER
021404  012700  001604                  MOV     #15.*60.,R0             ; GET REPORT INTERVAL
021410  004737  017614                  CALL    SETTO                   ; SET TIME FOR NEXT REPORT

021414                          RPTXX:  POP     <R5,R4,R3,R2,R1,R0>

021430                                  EXIT    RPT
021430  000167                          .WORD   J$JMP
021432  000034                          .WORD   L10037-2-.

021434     116      042      124 RPTMSG: .ASCIZ \N"TEST "D3" IN PROGRESS.  "\
                                        .EVEN

                                        ENDRPT
021470                          L10037:
021470  104425                          TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

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

021472                                  BGNPROT
021472                          L$PROT::

021472  177777                          -1                      ; P-TABLE OFFSET FOR CSR ADDRESS
021474  177777                          -1                      ; P-TABLE OFFSET FOR MASSBUS ADDRESS
021476  177777                          -1                      ; P-TABLE OFFSET FOR DRIVE NUMBER

021500                                  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
                                L$INIT::
                                                                        ;HERE FROM START COMMAND?
                                        READEF  #EF.STA
021500  012700  000040                  MOV     #EF.STA,R0
021504  104447                          TRAP    C$REFG
                                                                        ;BRANCH TO 1$ IF NOT, ELSE
                                        BNCOMPLETE      1$
021506  103004                          BCC     1$
021510  012737  000010  002202          MOV     #ISTRT,IFLAGS           ;SET START BIT IN FLAG.
021516  000525                          BR      INIT1
                                1$:                                     ;HERE FROM RESTART COMMAND?
                                        READEF  #EF.RES
021520  012700  000037                  MOV     #EF.RES,R0
```

```
    021524  104447                          TRAP    C$REFG
                                                                    ;BRANCH TO 2$ IF NOT, ELSE
53
54 021526                                   BNCOMPLETE      2$
   021526  103004                           BCC     2$
55 021530  052737  000004  002202          BIS     #IREST,IFLAGS       ;SET RESTART BIT IN FLAG.
56 021536  000515                          BR      INIT1
57 021540                           2$:                                 ;HERE FROM CONTINUE COMMAND?
58 021540                                   READEF  #EF.CON
   021540  012700  000036                   MOV     #EF.CON,R0
   021544  104447                           TRAP    C$REFG
                                                                    ;BRANCH TO 3$ IF NOT, ELSE
59
60 021546                                   BNCOMPLETE      3$
   021546  103007                           BCC     3$
61 021550  042737  000020  002202          BIC     #ISTRTH,IFLAGS      ;CLEAR 1ST TIME THRU CZUDIO FLAG AND
62 021556  052737  000002  002202          BIS     #ICONT,IFLAGS       ;SET CONTINUE BIT IN FLAG.
63 021564  000472                          BR      13$
64 021566                           3$:                                 ;HERE FROM POWER FAIL?
65 021566                                   READEF  #EF.PWR
   021566  012700  000034                   MOV     #EF.PWR,R0
   021572  104447                           TRAP    C$REFG
                                                                    ;BRANCH TO 4$ IF NOT, ELSE
66
67 021574                                   BNCOMPLETE      4$
   021574  103001                           BCC     4$
68 021576  000465                          BR      13$
69
70                                  ;
71                                  ;       MAKE ALL CONTROLLER/DRIVE TABLES NOT AVAILABLE FOR TESTING
72                                  ;
73
74 021600  013705  002172          4$:     MOV     CTABS,R5            ;GET ADDRESS OF 1ST CONTROLLER TABLE
75 021604  052765  100000  000002  5$:     BIS     #CT.AVL,C.UNIT(R5)  ;SET CONTROLLER TABLE NOT AVAILABLE
76 021612  010502                          MOV     R5,R2               ;GET POINTER TO DRIVE TABLES
77 021614  062702  000016                  ADD     #C.DRO,R2
78 021620  012703  000004                  MOV     #4.,R3              ; GET NUMBER OF DRIVES PER CONTROLLER
79 021624  012200                  6$:     MOV     (R2)+,R0            ;SEE IF THIS DRIVES HAS A TABLE.
80 021626  001403                          BEQ     7$                  ;BRANCH IF NOT, ELSE
81 021630  052760  100000  000002          BIS     #DT.AVL,D.UNIT(R0)  ;SET DRIVE TABLE NOT AVAILABLE.
82 021636  005303                  7$:     DEC     R3                  ;LOOK AT NEXT DRIVE IN CONTROLLER TABLE.
83 021640  001371                          BNE     6$                  ;BRANCH IF NO DRIVES, ELSE
84 021642  062705  000034                  ADD     #C.SIZE,R5          ;MOVE TO NEXT CONTROLLER TABLE
85 021646  005715                          TST     (R5)                ;IS THERE A NEXT ONE?
86 021650  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 021652  005003                          CLR     R3                  ;START WITH LOGICAL UNIT 0
92 021654                           8$:                                 ;GET POINTER TO IT'S P-TABLE
93 021654                                   GPHARD  R3,R0
   021654  010300                           MOV     R3,R0
   021656  104442                           TRAP    C$GPHRD
                                                                    ;BRANCH TO 12$ IF NOT AVAILABLE
94
95 021660                                   BNCOMPLETE      12$
   021660  103030                           BCC     12$
96 021662  013705  002172                  MOV     CTABS,R5            ;GET ADDRESS OF 1ST CONTROLLER TABLE
97 021666  021015                  9$:     CMP     (R0),(R5)           ;SEE IF CSR ADDRESSES ARE THE SAME.
98 021670  001411                          BEQ     11$                 ;BRANCH IF SO, ELSE
```

```
 99 021672  062705  000034                ADD     #C.SIZE,R5              ;LOOK AT NEXT CONTROLLER TABLE.
100 021676  005715                        TST     (R5)                    ;SEE IF THERE IS ANOTHER CONTROLLER TABLE.
101 021700  001372                        BNE     9$                      ;BRANCH IF SO. ELSE
102 021702                          10$:                                  ;REPORT TABLE CONSISTANCY ERROR.
103 021702                                ERRSF   6.,ERR006
    021702  104454                       TRAP    C$ERSF
    021704  000006                       .WORD   6
    021706  000000                       .WORD   0
    021710  011066                       .WORD   ERR006
104                                                                       ;DO CLEAN-UP TRAP
105 021712                                DOCLN
    021712  104444                       TRAP    C$DCLN
106
107 021714  016001  000006          11$:  MOV     H.DRV(R0),R1            ;GET DRIVE NUMBER FROM P-TABLE
108 021720  004737  015404                CALL    GTDRVT                  ;FIND THE DRIVE TABLE ADDRESS
109 021724  001366                        BNE     10$                     ;BRANCH IF NOT FOUND. ELSE
110 021726  042765  100000  000002        BIC     #CT.AVL,C.UNIT(R5)      ;CLEAR AVAILABLE BIT IN CONTROLLER AND
111 021734  042764  100000  000002        BIC     #DT.AVL,D.UNIT(R4)      ;THE DRIVE TABLES.
112 021742  005203                  12$:  INC     R3                      ;INCREMENT TO NEXT UNIT IN P-TABLE
113 021744  020337  002012                CMP     R3,L$UNIT               ;SEE IF ALL P-TABLES CHECKED.
114 021750  002741                        BLT     8$                      ;BRANCH IF NOT, ELSE
115 021752  012701  002272          13$:  MOV     #STIME,R1               ; GET REPORT TIMER
116 021756  012700  001604                MOV     #15.*60.,R0             ; GET REPORT INTERVAL
117 021762  004737  017614                CALL    SETTO                   ; SET TIME FOR NEXT REPORT
118 021766  000137  022574                JMP     INITXX                  ;EXIT THE INITIALIZE SECTION.
119
120                                 ;
121                                 ;       INITIALIZE KW11 CLOCK, FREE MEMORY AND IP ADDRESS TABLE
122                                 ;       DURING START OR RESTART COMMAND ONLY
123                                 ;
124
125 021772  005037  002266          INIT1:  CLR     KW.EL                   ;CLEAR ELAPSED TIME
126 021776  005037  002270                  CLR     KW.EL+2
127 022002                                  CLOCK   L,R0                    ;SEE IF L-CLOCK PRESENT
    022002  012700  000114                  MOV     #'L,R0
    022006  104462                         TRAP    C$CLCK
128 022010                                  BCOMPLETE       1$
    022010  103413                         BCS     1$
129 022012                                  CLOCK   P,R0                    ;SEE IF P-CLOCK PRESENT
    022012  012700  000120                  MOV     #'P,R0
    022016  104462                         TRAP    C$CLCK
130 022020                                  BCOMPLETE       1$
    022020  103407                         BCS     1$
131 022022  005037  002256                  CLR     KW.CSR                  ;IF NEITHER, CLEAR CSR STORAGE WORD
132 022026                                  PNTF    NOCLOCK
133 022036  000434                         BR      2$
134
135 022040  012037  002256          1$:     MOV     (R0)+,KW.CSR            ;STORE DATA RETURNED
136 022044  012037  002260                  MOV     (R0)+,KW.BRL
137 022050  012037  002262                  MOV     (R0)+,KW.VEC
138 022054  012037  002264                  MOV     (R0)+,KW.HZ
139                                                                       ;SETUP KW11 VECTOR ADDRESS
140 022060                                  SETVEC  KW.VEC,#KW11I,#PRI07
    022060  012746  000340                  MOV     #PRI07,-(SP)
    022064  012746  017572                  MOV     #KW11I,-(SP)
    022070  013746  002262                  MOV     KW.VEC,-(SP)
    022074  012746  000003                  MOV     #3,-(SP)
```

INITIALIZE SECTION

```
                022100  104437                          TRAP    C$SVEC
                022102  062706  000010                  ADD     #10,SP
    141         022106  012777  000105  160142          MOV     #KW.OUT,@KW.CSR         ;START THE CLOCK
    142         022114  012701  002272                  MOV     #STIME,R1               ; GET REPORT TIMER
    143         022120  012700  001604                  MOV     #15.*60.,R0             ; GET REPORT INTERVAL
    144         022124  004737  017614                  CALL    SETTO                   ; SET TIME FOR NEXT REPORT
    145         022130  004737  020744          2$:     CALL    RESET                   ;RESET ALL CONTROLLERS
    146         022134                                  MEMORY  FFREE                   ;RESET START OF FREE MEMORY
                022134  104431                          TRAP    C$MEM
                022136  010037  002160                  MOV     R0,FFREE
    147         022142  017737  160012  002162          MOV     @FFREE,FSIZE            ;RESET SIZE OF FREE MEMORY
    148         022150  005037  002204                  CLR     TNUM                    ;INITIALIZE TEST NUMBER TO NO TEST RUNNING
    149
    150                                         ;
    151                                         ;       ALLOCATE DRIVE TABLES TO MEMORY
    152                                         ;
    153
    154         022154  013737  002160  002170  INIT2:  MOV     FFREE,DTABS             ;STORE START OF DRIVE TABLES AND
    155         022162  005077  160002                  CLR     @DTABS                  ;MARK ZERO END.
    156         022166  013700  002012                  MOV     L$UNIT,R0               ;GET NUMBER OF LOGICAL UNITS TO RUN,
    157         022172  012701  000001                  MOV     #1,R1                   ;GET INITIAL SIZE OF DRIVE TABLE AND
    158         022176  062701  000002          1$:     ADD     #<D.SIZE>/2,R1          ;ACCUMULATE DRIVE TABLE SIZE.
    159         022202  005300                          DEC     R0                      ;SEE IF ANY MORE LOGICAL UNITS.
    160         022204  001374                          BNE     1$                      ;BRANCH IF NOT, ELSE
    161         022206  004737  012132                  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         022212  013737  002160  002172  INIT3:  MOV     FFREE,CTABS             ; STORE START OF CONTROLLER TABLES AND
    169         022220  005077  157746                  CLR     @CTABS                  ; MARK ZEROS END.
    170         022224  005037  002174                  CLR     CTRLRS                  ; CLEAR CONTROLLER COUNT
    171         022230  012701  002246                  MOV     #IPADRS,R1              ; R1 -> IP ADDRESS
    172         022234  012702  000004                  MOV     #4.,R2                  ; GET MAXIMUM # OF CONTROLLERS
    173         022240  005021                  1$:     CLR     (R1)+                   ; CLEAR ENTRY
    174         022242  005302                          DEC     R2                      ; DONE?
    175         022244  001375                          BNE     1$                      ; IF NOT, BRANCH
    176
    177                                         ;
    178                                         ;       BUILD CONTROLLER TABLES
    179                                         ;
    180
    181         022246  005005                  INIT4:  CLR     R5                      ;CLEAR CUSTOMER DATA FLAG
    182         022250  005002                          CLR     R2                      ;START WITH LOGICAL UNIT 0
    184         022252                          1$:     GPHARD  R2,R0                   ;GET POINTER TO IT'S P-TABLE
                022252  010200                          MOV     R2,R0
                022254  104442                          TRAP    C$GPHRD
    185         022256                                  BNCOMPLETE      16$             ;BRANCH TO 16$ IF NOT AVAILABLE
                022256  103130                          BCC     16$
    186         022260  013703  002172                  MOV     CTABS,R3                ;GET ADDRESS OF 1ST CONTROLLER TABLE
    187         022264  005713                  2$:     TST     (R3)                    ;CHECK IF ANY MORE TABLES
    188         022266  001431                          BEQ     6$                      ;BUILD NEW TABLE IF FOUND ZERO WORD
    189         022270  021013                          CMP     (R0),(R3)               ;CHECK IF SAME CSR ADDRESS.
    190         022272                                          ASSUME  C.UADR EQ 0
    191         022272                                          ASSUME  H.UBA EQ 0
```

```
                        001013                          BNE     4$                      ;BRANCH IF NOT, ELSE
                                                                                        ;CHECK THAT OTHER PARAMETERS MATCH.
                        016004  000004                  MOV     H.BRL(R0),R4            ;GET BR LEVEL FROM P-TABLE
                        000304                          SWAB    R4                      ;SWAP TO HIGH BYTE
                        006104                          ROL     R4                      ;SHIFT ONE MORE TO LEFT
                        056004  000002                  BIS     H.VEC(R0),R4            ;ADD VECTOR ADDRESS
                        020463  000004                  CMP     R4,C.VEC(R3)            ;COMPARE VECTOR AND BR LEVELS.
                        001457                          BEQ     11$                     ;BRANCH IF SAME, ELSE
                        000137  022606          3$:     JMP     CTABER                  ;FOUND SAME CONTROLLER WITH DIFFERENT
                                                                                        ;  BR LEVEL OR VECTOR ADDRS
                        016304  000004          4$:     MOV     C.VEC(R3),R4            ;GET VECTOR FROM CONTROLLER TABLE
                        042704  177000                  BIC     #^C<CT.VEC>,R4          ;AND
                        026004  000002                  CMP     H.VEC(R0),R4            ;COMPARE VECTOR ADDRESSES.
                        001002                          BNE     5$                      ;BRANCH IF DIFFERENT, ELSE
                        000137  022656                  JMP     SAMVEC                  ;FOUND TWO CONTROLLERS WITH SAME VECTOR

                        062703  000034          5$:     ADD     #C.SIZE,R3              ;POINT TO BEGINNING OF NEXT CONTROLLER
                        000745                          BR      2$                      ;TABLE IN MEMORY.


                                                ;
                                                ;       BUILD NEW CONTROLLER TABLE
                                                ;

                        012704  002246          6$:     MOV     #IPADRS,R4              ;GET BEGINNING OF IP ADDRESS TABLE
                        020427  002256          7$:     CMP     R4,#IPADRS+8.           ;SEE IF END OF IP ADDRESS TABLE.
                        101004                          BHI     9$                      ;BRANCH IF SO, ELSE
                        005724                          TST     (R4)+                   ;DID WE FIND AN OPEN ENTRY ?
                        001401                          BEQ     8$                      ;BRANCH IF SO, ELSE
                        000772                          BR      7$                      ;LOOK AGAIN.

                        011044          8$:     MOV     (R0),-(R4)              ;TAKE CSR ADDRESS FROM P-TABLE
                                                                                ;AND STORE IT IN THE IP ADDRESS TABLE.
                        012701  000016  9$:     MOV     #<C.SIZE>/2,R1          ;GET # OF ENTRIES IN CONTROLLER TABLE
                        004737  012132          CALL    ALOCM                   ;AND ALLOCATE A TABLE TO MEMORY.
                                                                                ; R0 => 1ST WORD P-TABLE
                                                                                ; R1 => 1ST WORD IN CONTROLLER TABLE
                        011021                  MOV     (R0),(R1)+              ; STORE CSR ADDRESS AND
                        010221                  MOV     R2,(R1)+                ; UNIT NUMBER IN THE CONTROLLER TABLE.
                        016004  000004          MOV     H.BRL(R0),R4            ;GET THE BR LEVEL.
                        000304                  SWAB    R4                      ;SWAP TO HIGH BYTE.
                        006104                  ROL     R4                      ;SHIFT ONE MORE TO LEFT.
                        056004  000002          BIS     H.VEC(R0),R4            ;ADD VECTOR ADDRESS AND
                        010421                  MOV     R4,(R1)+                ;STORE IT IN THE CONTROLLER TABLE.
                        012721  004037          MOV     #4037,(R1)+             ;THE 'JSR R0' INSTRUCION AND
                        012721  017554          MOV     #CNTSRV,(R1)+           ;THE ADDRESS OF THE INTERRUPT SERVICE
                                                                                ;ROUTINE IN THE CONTROLLER TABLE.
                        012704  000011          MOV     #<C.SIZE-C.FLG>/2,R4    ;GET # OF ENTRIES TO END OF TABLE.
                        005021          10$:    CLR     (R1)+                   ;CLEAR REST OF TABLE AND
                        005304                  DEC     R4                      ;ADD ZERO WORD AT END.
                        002375                  BGE     10$                     ;LOOP TIL ALL CLEARED
                        005237  002174          INC     CTRLRS                  ;KEEP TRACK OF CONTROLLER COUNT


                                                ;
                                                ;       BUILD DRIVE TABLES
                                                ;

                        013701  002170  11$:    MOV     DTABS,R1                ;GET ADDRESS OF CURRENT DRIVE TABLE
```

```
    267 022460  062703  000016                ADD     #C.DRO,R3               ; INDEX TO 1ST DRIVE IN TABLE
    268 022464  012704  000004                MOV     #4.,R4                  ; GET # OF DRIVES PER CONTROLLER
    269 022470  005713                  12$:  TST     (R3)                    ;ANY ENTRY TO DRIVE TABLE.
    270 022472  001411                        BEQ     14$                     ;BRANCH IF NOT, ELSE
    271 022474  026033  000006                CMP     H.DRV(R0),@(R3)+        ;COMPARE DRIVE NUMBER IN DRIVE TABLE.
    272 022500  001002                        BNE     13$                     ;BRANCH IF DIFFERNENT, ELSE
    273 022502  000137  022622                JMP     MLDRER                  ;FOUND TWO P-TABLES WITH SAME DRIVE.
    274
    275 022506  005304                  13$:  DEC     R4                      ; COUNT DRIVES
    276 022510  001367                        BNE     12$                     ; IF FOUR DRIVE TABLES ALREADY EXIST,
    277 022512  000137  022640                JMP     TOOMER                  ;  THEN REPORT ERROR
    278
    279 022516  010113                  14$:  MOV     R1,(R3)                 ;STORE ADDRESS OF DRIVE TABLE IN
    280                                                                       ;CONTROLLER TABLE.
    281 022520  016021  000006                MOV     H.DRV(R0),(R1)+         ;STORE DRIVE NUMBER AND
    282 022524  010221                        MOV     R2,(R1)+                ;LOGICAL UNIT NUMBER IN DRIVE TABLE.
    295
    296 022526  062737  000004  002170        ADD     #D.SIZE,DTABS           ;NEXT DRIVE TABLE ADDRESS AND
    297 022534  005077  157430                CLR     @DTABS                  ;MARK ZERO END.
    298 022540  005202                  16$:  INC     R2                      ;INCREMENT LOGICAL UNIT NUMBER
    299 022542  020237  002012                CMP     R2,L$UNIT               ;CHECK IF GOT ALL TABLES
    300 022546  002641                        BLT     1$                      ;IF NOT, GO BACK FOR NEXT, ELSE
    301 022550  012701  000001                MOV     #1,R1                   ;GET 1 WORD TO TERMINATE ALL CONTROLLER
    302 022554  004737  012132                CALL    ALOCM                   ;TABLES AND ALLOCATE IT TO MEMORY.
    303
    340                                     ;
    341                                     ;       SAVE CURRENT PARAMETERS TO FREE MEMORY SO EACH TEST CAN USE ALL OF IT
    342                                     ;
    343
    344 022560  013737  002160  002164  INIT6:  MOV     FFREE,FMEM              ;SAVE START ADDRESS
    345 022566  013737  002162  002166          MOV     FSIZE,FMEMS             ;SAVE SIZE
    346
    347                                     ;
    348                                     ;       EXIT INITIALIZE SECTION
    349                                     ;
    350
    351 022574                        INITXX: SETPRI  #PRI00                  ;SET RUNNING PRIORITY TO ZERO
        022574  012700  000000                MOV     #PRI00,R0
        022600  104441                        TRAP    C$SPRI
    352
    353 022602                                EXIT    INIT
        022602  104432                        TRAP    C$EXIT
        022604  000066                        .WORD   L10041-.
```

```
                                .SBTTL  INITIALIZE ERRORS

                                ;
                                ;       DIFFERENT VECTORS OR BR LEVELS FOR ONE CONTROLLER
                                ;

022606  010305                  CTABER: MOV     R3,R5                   ;GET CONTROLLER ADDRESS
022610                                  ERRSF 1,,ERR001
022610  104454                          TRAP    C$ERSF
022612  000001                          .WORD   1
022614  000000                          .WORD   0
022616  011002                          .WORD   ERR001
                                                                        ;DO CLEAN-UP TRAP
022620                                  DOCLN
022620  104444                          TRAP    C$DCLN

                                ;
                                ;       TWO P-TABLES FOR SAME DRIVE
                                ;

022622  013705  002216          MLDRER: MOV     TEMP,R5                 ;GET CONTROLLER ADDRESS
022626                                  ERRSF 2,,ERR002
022626  104454                          TRAP    C$ERSF
022630  000002                          .WORD   2
022632  000000                          .WORD   0
022634  011020                          .WORD   ERR002
                                                                        ;DO CLEAN-UP TRAP
022636                                  DOCLN
022636  104444                          TRAP    C$DCLN

                                ;
                                ;       MORE THAN FOUR DRIVES SELECTED ON ONE CONTROLLER
                                ;

022640  013705  002216          TOOMER: MOV     TEMP,R5                 ;GET CONTROLLER ADDRESS
022644                                  ERRSF 3,,ERR003
022644  104454                          TRAP    C$ERSF
022646  000003                          .WORD   3
022650  000000                          .WORD   0
022652  011036                          .WORD   ERR003
                                                                        ;DO CLEAN-UP TRAP
022654                                  DOCLN
022654  104444                          TRAP    C$DCLN

                                ;
                                ;       TWO CONTROLLERS USE THE SAME VECTOR
                                ;

022656  010305                  SAMVEC: MOV     R3,R5                   ;GET CONTROLLER ADDRESS
022660                                  ERRSF   8,,ERR008
022660  104454                          TRAP    C$ERSF
022662  000010                          .WORD   8
022664  000000                          .WORD   0
022666  011100                          .WORD   ERR008
                                                                        ;DO CLEAN-UP TRAP
022670                                  DOCLN
022670  104444                          TRAP    C$DCLN
```

INITIALIZE ERRORS

```
     42 022672                              ENDINIT
        022672                      L10041:
        022672  104411                      TRAP    C$INIT
     43
```

```
                                .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.
                                ;--

022674                                  BGNAUTO
022674                          L$AUTO::

022674                                  ENDAUTO
022674                          L10042:
022674  104461                          TRAP    C$AUTO
```

```
                                .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".
                                ;--

022676                                  BGNCLN
022676                          L$CLEAN::
022676  004737  020744                  CALL    RESET                   ;RESET ALL CONTROLLERS

022702                                  EXIT    CLN
022702  104432                          TRAP    C$EXIT
022704  000002                          .WORD   L10043-.

022706                                  ENDCLN
022706                          L10043:
022706  104412                          TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

 022710                                 BGNDU
 022710                         L$DU::

 022710                                 EXIT    DU
 022710  000167                         .WORD   J$JMP
 022712  000000                         .WORD   L10044-2-.

 022714                                 ENDDU
 022714                         L10044:
 022714  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.
                                ;--

022716                                  BGNAU
022716                          L$AU::

022716                                  EXIT    AU
022716  000167                          .WORD   J$JMP
022720  000000                          .WORD   L10045-2-.

022722                                  ENDAU
022722                          L10045:
022722  104452                          TRAP    C$AU
```

```
                                .SBTTL HARDWARE TESTS
                                
                                .SBTTL  TEST 1: BUS ADDRESSING TEST
                                
                                ;**
                                ;       TEST 1 - BUS ADDRESSING TEST
                                ;--
                                
                                        BGNTST
022724                          T1::
                                
022724  012701  000001                MOV     #1,R1                   ; INITIALIZE TEST PARAMETERS
022730  004737  013040                CALL    TINIT                   ;
022734  013737  002172  002176        MOV     CTABS,TSTTAB            ; GET ADDRESS OF 1ST CONTROLLER TABLE
022742  013705  002176        T1NEXT: MOV     TSTTAB,R5               ; GET CONTROLLER TABLE ADDRESS
022746  116537  000002  002074        MOVB    C.UNIT(R5),L$LUN        ; CHECK IF UNIT AVAILABLE FOR TESTING
022754  005765  000002                TST     C.UNIT(R5)              ;
022760  100010                        BPL     T1NOW                   ; TEST IF AVAILABLE
022762                                ASSUME CT.AVL EQ BIT15
022762  062737  000034  002176  T1SKIP: ADD     #C.SIZE,TSTTAB          ; GET NEXT CONTROLLER TABLE ADDRESS
022770  005777  157202                TST     @TSTTAB                 ; CHECK IF ANOTHER CONTROLLER TABLE
022774  001362                        BNE     T1NEXT                  ; DO NEXT CONTROLLER
022776                                EXIT TST
022776  104432                        TRAP    C$EXIT
023000  001522                        .WORD   L10046-.
                                
023002  004737  020744        T1NOW:  CALL    RESET                   ; RESET ALL CONTROLLERS
                                
```

```
                                        ;
                                        ;       TEST 1, SUBTEST 1
                                        ;       HOST MEMORY I/O PAGE ADDRESSING TEST
                                        ;
                                        BGNSUB: 1
                                        T1.1:
023006  104402                                  TRAP    C$BSUB
023010  005037  002276                          CLR     NXMAD                   ; CLEAR MEMORY ERROR FLAG
023014                                          SETVEC  #ERRVEC,#NXMI,#PRI07    ; SETUP TIMEOUT ERROR VECTOR
023014  012746  000340                          MOV     #PRI07,-(SP)
023020  012746  017544                          MOV     #NXMI,-(SP)
023024  012746  000004                          MOV     #ERRVEC,-(SP)
023030  012746  000003                          MOV     #3,-(SP)
023034  104437                                  TRAP    C$SVEC
023036  062706  000010                          ADD     #10,SP
023042  011504                                  MOV     (R5),R4                 ; GET ADDRESS OF IP REGISTER
023044  005714                                  TST     (R4)                    ; READ IP REGISTER
023046  005764  000002                          TST     2(R4)                   ; READ SA REGISTER
023052                                          CLRVEC  #ERRVEC                 ; RETURN TIMEOUT ERROR VECTOR
023052  012700  000004                          MOV     #ERRVEC,R0
023056  104436                                  TRAP    C$CVEC
023060  005737  002276                          TST     NXMAD                   ; CHECK FLAG
023064  001406                                  BEQ     1$                      ; IF OK, GO TO NEXT SUBTEST
023066                                          ERRDF 38,,ERR038                ; ERROR - MEMORY ADDRESS ERROR
023066  104455                                  TRAP    C$ERDF
023070  000046                                  .WORD   38
023072  000000                                  .WORD   0
023074  011532                                  .WORD   ERR038
023076                                          CKLOOP                          ; LOOP ON ERROR IF SELECTED
023076  104406                                  TRAP    C$CLP1
023100  000730                                  BR      T1SKIP                  ; END TEST NOW
023102                                  1$:
023102                                  ENDSUB
023102                                  L10047:
023102  104403                                  TRAP    C$ESUB
```

```
      1
      2                                 ;
      3                                 ;       TEST1, SUBTEST 2
      4                                 ;       MAKE SURE CONTROLLER PASSES INTEPNAL DIAGNOSTIC
      5                                 ;       MAKE SURE CONTROLLER CAN SENSE STEP 1 AND 2
      6                                 ;
      7
      8 023104                          BGNSUB: 2
        023104                          T1.2:
        023104  104402                          TRAP    C$BSUB
      9 023106  005014                          CLR     (R4)                    ; INIT CONTROLLER
     10 023110  012737  004000  002330          MOV     #SA.S1,CNTRSD           ; STEP 1 ASSERTED?
     11 023116  004737  020566                  CALL    CNTRSP                  ; WAIT FOR RESPONSE
     12 023122  103717                          BCS     T1SKIP                  ; IF ERROR, EXIT
     13 023124  012764  100000  000002          MOV     #SA.STP,2(R4)           ; SEND STEP 1
     14 023132  012737  010000  002330          MOV     #SA.S2,CNTRSD           ; STEP 2 ASSERTED?
     15 023140  004737  020566                  CALL    CNTRSP                  ; WAIT FOR RESPONSE
     16 023144  103706                          BCS     T1SKIP                  ; IF ERROR, EXIT
     17 023146                          ENDSUB
        023146                          L10050:
        023146  104403                          TRAP    C$ESUB
     18
```

```
      1
      2                                 ;
      3                                 ;       TEST 1, SUBTEST 3
      4                                 ;       TEST THE DIAGNOSTIC WRAP MODE OF ALL CONTROLLERS ON THE SYSTEM
      5                                 ;
      6
      7 023150                          BGNSUB; 3
        023150                          T1.3:
        023150  104402                          TRAP    C$BSUB
      8 023152  011504                          MOV     (R5),R4                 ; R4 POINTS TO IP REGISTER
      9 023154                                  ASSUME C.UADR EQ 0
     10 023154  005014                          CLR     (R4)                    ; INITIALIZE THE CONTROLLER
     11 023156  012737  004000  002330          MOV     #SA.S1,CNTRSD           ; LOOK FOR STEP 1
     12 023164  004737  020566                  CALL    CNTRSP                  ; WAIT FOR RESPONSE
     13 023170  103674                          BCS     T1SKIP                  ; IF ERROR, BRANCH
     14 023172  012737  140000  002332          MOV     #<SA.STP+SA.WRP>,WCHNGD ; SAVE NEW PORT CONTENTS
     15 023200  012764  140000  000002          MOV     #<SA.STP+SA.WRP>,2(R4)  ; INITIALIZE FOR PORT WRAP
     16 023206  004737  021240                  CALL    WCHNG                   ; WAIT FOR THE PORT TO CHANGE
     17 023212  001263                          BNE     T1SKIP                  ; IF ERROR, BRANCH
     18                                 ;       CMP     #<SA.STP+SA.WRP>,2(R4)  ;[TEMP] COMPARE WITH DATA WRITTEN
     19                                 ;       BNE     3$                      ;[TEMP]
     20 023214  012702  000001          1$:     MOV     #1,R2                   ; SET UP FOR SHIFTING '1'
     21 023220  012703  000020                  MOV     #16.,R3                 ; SET UP LOOP COUNT
     22 023224  010237  002332          2$:     MOV     R2,WCHNGD               ; SAVE NEW PORT CONTENTS
     23 023230  010264  000002                  MOV     R2,2(R4)                ; WRITE PATTERN TO SA FOR LOOP
     24 023234  004737  021240                  CALL    WCHNG                   ; WAIT FOR SA TO CHANGE
     25 023240  001250                          BNE     T1SKIP                  ; IF ERROR, BRANCH
     26                                 ;       CMP     R2,2(R4)                ;[TEMP] COMPARE RO WITH ECHOED DATA
     27                                 ;       BEQ     4$                      ;[TEMP] IF MATCH, BRANCH
     28 023242  000405                          BR      4$                      ;[TEMP]
     29 023244                          3$:     ERRDF   26.,ERR026              ; REPORT ERROR
        023244  104455                          TRAP    C$ERDF
        023246  000032                          .WORD   26
        023250  000000                          .WORD   0
        023252  011344                          .WORD   ERR026
     30 023254  000642                          BR      T1SKIP                  ; BRANCH
     31 023256  006302                  4$:     ASL     R2                      ; MOVE THE SHIFTING ONE LEFT BY 1
     32 023260  005303                          DEC     R3                      ; DECREMENT COUNT
     33 023262  001360                          BNE     2$                      ; IF LOOP INCOMPLETE, BRANCH
     34 023264                          ENDSUB
        023264                          L10051:
        023264  104403                          TRAP    C$ESUB
     35
```

```
      1
      2                                 ;
      3                                 ;       TEST1, SUBTEST 4
      4                                 ;       INITIALIZE CONTROLLER WITH SMALLEST RING BUFFER AND INTERRUPTS
      5                                 ;       DISABLED
      6                                 ;
      7
      8 023266                          BGNSUB: 4
        023266                          T1.4:
        023266  104402                          TRAP    C$BSUB
      9 023270  005004                          CLR     R4                      ; LOAD INIT CHARACTERISTICS IN R4
     10 023272  004737  017676                  CALL    CNTINT                  ; INIT CONTROLLER
     11 023276  001631                          BEQ     T1SKIP                  ; IF ERROR, EXIT TEST
     12 023300                          ENDSUB
        023300                          L10052:
        023300  104403                          TRAP    C$ESUB
     13
```

TEST 1: BUS ADDRESSING TEST

```
      1
      2                                 ;
      3                                 ;       TEST 1, SUBTEST 5
      4                                 ;       TEST THE INTERRUPTS VECTOR AND BR LEVEL
      5                                 ;
      6
      7 023302                          BGNSUB: 5
        023302                          T1.5:
        023302  104402                           TRAP    C$BSUB
      8 023304  011504                           MOV     (R5),R4                 ; R4 POINTS TO IP REGISTER
      9 023306                                    ASSUME C.UADR EQ 0
     10 023306  016503  000004                    MOV     C.VEC(R5),R3            ; GET VECTOR AND BR LEVEL
     11 023312  010302                           MOV     R3,R2                   ; COPY TO R2 FOR BR LEVEL
     12 023314  042703  177000                    BIC     #^CCT.VEC,R3            ; CLEAR UNUSED VECTOR BITS
     13 023320  042702  170777                    BIC     #^CCT.BRL,R2            ; CLEAR UNUSED BRANCH LEVEL BITS
     14 023324  000302                           SWAB    R2                      ; GET BR LEVEL IN LOW BYTE
     15 023326  006202                           ASR     R2                      ; DIVIDE BY 2
     16 023330  010237  002334                    MOV     R2,BRLEV                ; SAVE THE BUS REQUEST LEVEL
     17 023334                                    PNTX    INTST0,(R5),R3          ; PRINT BEGINNING OF INTERRUPT MESSAGE
     18 023350                                    ASSUME C.UADR EQ 0
     19 023350                                    SETVEC  R3,#INTSRV,#PRI00       ; SETUP INTERRUPT VECTOR ADDRESS
        023350  012746  000000                    MOV     #PRI00,-(SP)
        023354  012746  017564                    MOV     #INTSRV,-(SP)
        023360  010346                           MOV     R3,-(SP)
        023362  012746  000003                    MOV     #3,-(SP)
        023366  104437                           TRAP    C$SVEC
        023370  062706  000010                    ADD     #10,SP
     20 023374                                    SETPRI  #PRI00                  ; SET PRIORITY = 0 TO CHECK INTERRUPTS
        023374  012700  000000                    MOV     #PRI00,R0
        023400  104441                           TRAP    C$SPRI
     21 023402  006203                           ASR     R3                      ; DIVIDE VECTOR BY 4 FOR
     22 023404  006203                           ASR     R3                      ;               CONTROLLER INITIALIZATION
     23 023406  052703  100200                    BIS     #<SA.STP+SA.INT>,R3     ; SET OTHER BITS FOR INITIALIZATION
     24 023412  005037  002214                    CLR     INTRCV                  ; FLAG AS NO INTERRUPTS RECEIVED
     25 023416  005014                           CLR     (R4)                    ; INIT CONTROLLER
     26 023420  012737  004000  002330           MOV     #SA.S1,CNTRSD           ; LOOK FOR STEP 1 COMPLETION
     27 023426  004737  020566                    CALL    CNTRSP                  ; WAIT FOR COMPLETION
     28 023432  103514                           BCS     9$                      ; IF ERROR, EXIT
     29 023434  010364  000002                    MOV     R3,2(R4)                ; MOVE STEP 1 DATA TO CONTROLLER
     30 023440  012700  000012                    MOV     #10.,R0                 ; SET UP TIMEOUT OF 10 SECONDS
     31 023444  010501                           MOV     R5,R1                   ; POINT TO CONTROLLER TABLE
     32 023446  062701  000026                    ADD     #C.TO,R1                ; MOVE POINTER TO TIMEOUT FIELD
     33 023452  004737  017614                    CALL    SETTO                   ; INITIALIZE TIMER
     34 023456  005737  002214          2$:      TST     INTRCV                  ; SEE IF INTERRUPTED
     35 023462  001016                           BNE     3$                      ; IF SO, EVERYTHING'S OK, SO BRANCH
     36 023464                                    BREAK                           ; >>>>>>>BREAK BACK TO MONITOR<<<<<<<
        023464  104422                           TRAP    C$BRK
     37
     38 023466  005737  002256                    TST     KW.CSR                  ; SEE IF CLOCK ON SYSTEM
     39 023472  001771                           BEQ     2$                      ; IF NOT, DON'T TIMEOUT
     40 023474  023765  002270  000030           CMP     KW.EL+2,C.TOH(R5)       ; CHECK HIGH WORD OF TIMER
     41 023502  101043                           BHI     6$                      ; IF GREATER, TIMER EXPIRED
     42 023504  001364                           BNE     2$                      ; IF NOT SAME, KEEP WAITING
     43 023506  023765  002266  000026           CMP     KW.EL,C.TO(R5)          ; CHECK LOW WORD OF TIMER
     44 023514  103760                           BLO     2$                      ; IF LESS, KEEP WAITING
     45 023516  000435                           BR      6$                      ; ELSE, TIMER EXPIRED
     46
```

```
     023520  005037  002214          3$:     CLR     INTRCV                  ; FLAG AS NO INTERRUPTS RECEIVED
     023524                                  SETPRI  #PRI07                  ; SET PRIORITY AS HIGHEST PRIORITY
     023524  012700  000340                 MOV     #PRI07,R0
     023530  104441                         TRAP    C$SPRI
     023532  016402  000002                 MOV     2(R4),R2                ; READ SA REGISTER
     023536  005064  000002                 CLR     2(R4)                   ; WRITE SECOND STEP TO CONTROLLER
     023542  012702  000144                 MOV     #100.,R2                ; SET UP DELAY TO WAIT FOR INTERRUPT
     023546  005302                  4$:    DEC     R2                      ; DECREMENT COUNT
     023550  001376                         BNE     4$                      ; IF INCOMPLETE, BRANCH
     023552  012701  000007                 MOV     #7.,R1                  ; R1 IS PROCESS PRIORITY LEVEL
     023556                          5$:    PUSH    R1                      ; SAVE PRIORITY
     023560  006301                         ASL     R1                      ; SHIFT
     023562  006301                         ASL     R1                      ;     PRIORITY
     023564  006301                         ASL     R1                      ;             LEFT
     023566  006301                         ASL     R1                      ;                 5
     023570  006301                         ASL     R1                      ;                   BITS
     023572                                  SETPRI  R1                      ; SET RUNNING PRIORITY TO R1
     023572  010100                         MOV     R1,R0
     023574  104441                         TRAP    C$SPRI
     023576                                  POP     R1                      ; RESTORE R1
     023600  005737  002214                 TST     INTRCV                  ; SEE IF INTERRUPT RECEIVED
     023604  001007                         BNE     7$                      ; IF SO, BRANCH
     023606  005301                         DEC     R1                      ; DECREMENT PRIORITY LEVEL
     023610  100362                         BPL     5$                      ; IF ALL LEVELS UNTESTED, BRANCH
     023612                          6$:    ERRDF   28.,ERR028              ; REPORT NO INTERRUPTS ERROR
     023612  104455                         TRAP    C$ERDF
     023614  000034                         .WORD   28
     023616  000000                         .WORD   0
     023620  011402                         .WORD   ERR028
     023622  000420                         BR      9$                      ; BRANCH

     023624                          7$:    SETPRI  #PRI00                  ; SET RUNNING PRIORITY TO 0
     023624  012700  000000                 MOV     #PRI00,R0
     023630  104441                         TRAP    C$SPRI
     023632  005201                         INC     R1                      ; SO PRIORITY = BR LEVEL
     023634  013703  002326                 MOV     SSTEP4,R3               ; GET BR LEVEL SAVED IN CNTINT
     023640  001402                         BEQ     8$                      ; IF ZERO, THIS IS UDA50A - SKIP
     023642  010337  002334                 MOV     R3,BRLEV                ; KDA50-Q MUST USE DEFAULT BR LEVEL
     023646  023701  00233A          8$:    CMP     BRLEV,R1                ; SEE IF BR LEVEL MATCHES PRIORITY
     023652  001414                         BEQ     10$                     ; IF SO, BRANCH
     023654                                  ERRDF   29.,ERR029              ; REPORT ERROR
     023654  104455                         TRAP    C$ERDF
     023656  000035                         .WORD   29
     023660  000000                         .WORD   0
     023662  011414                         .WORD   ERR029
     023664  016503  000004          9$:    MOV     C.VEC(R5),R3            ; GET VECTOR ADDRESS
     023670  042703  177000                 BIC     #^CCT.VEC,R3            ; CLEAR UNUSED BITS
     023674                                  CLRVEC  R3                      ; CLEAR VECTOR
     023674  010300                         MOV     R3,R0
     023676  104436                         TRAP    C$CVEC
     023700  000137  022762                 JMP     T1SKIP                  ; EXIT TEST ON ERROR
     023704                          10$:   PNTX    INTST1                  ; PRINT TESTING COMPLETED
     023714  016503  000004                 MOV     C.VEC(R5),R3            ; GET VECTOR ADDRESS
     023720  042703  177000                 BIC     #^CCT.VEC,R3            ; CLEAR UNUSED BITS
     023724                                  CLRVEC  R3                      ; CLEAR VECTOR
     023724  010300                         MOV     R3,R0
     023726  104436                         TRAP    C$CVEC
```

```
     86 023730                          ENDSUB
        023730                          L10053:
        023730  104403                            TRAP    C$ESUB
     87
```

```
      1
      2                                 ;
      3                                 ;       TEST 1, SUBTEST 6
      4                                 ;       INITIALIZE CONTROLLER WITH LARGE RING BUFFER LARCE ENOUGH TO COVER
      5                                 ;       NORMAL HOST COMMUNICATIONS AREA PACKET AND BUFFER SPACE (A 5 IN
      6                                 ;       MESSAGE LENGTH AND A 5 IN COMMAND LENGTH).
      7                                 ;
      8
      9 023732                          BGNSUB: 6
        023732                          T1.6:
        023732  104402                           TRAP    C$BSUB
     10 023734  012704  126400                    MOV     #<SA.STP+<5*SA.MS1>+<5*SA.CM1>>,R4 ; LOAD CHARACTERISTICS
     11 023740  004737  017676                    CALL    CNTINT                  ; INIT CONTROLLER
     12 023744  001002                            BNE     1$                      ; IF NO ERROR, CONTINUE
     13 023746  000137  022762                    JMP     T1SKIP                  ; ELSE, EXIT TEST
     14 023752                          1$:
     15 023752                          ENDSUB
        023752                          L10054:
        023752  104403                           TRAP    C$ESUB
     16
```

```
      1
      2                                 ;**
      3                                 ;       TEST 1, SUBTEST 7
      4                                 ;       RUN DM PROGRAM IN CONTROLLER CURRENTLY UNDER TEST
      5                                 ;--
      6
      7 023754                          BGNSUB: 7
        023754                          T1.7:
        023754  104402                          TRAP    C$BSUB
      8 023756                                  PUSH    <FFREE,FSIZE>           ; SAVE ON STACK
      9 023766  012701  000001                  MOV     #1,R1                   ; SAVE COUNT OF CONTROLLERS TO RUN
     10 023772  004737  013104                  CALL    RUNDM                   ; RUN DM PROGRAM
     11 023776  001407                          BEQ     1$                      ; EXIT IF ERROR
     12 024000  004737  013200                  CALL    RESFDM                  ; WAIT FOR RESPONSE
     13 024004                                  POP     <FSIZE,FFREE>           ; RESTORE FROM STACK
     14 024014  000406                          BR      2$                      ;
     15 024016                          1$:     POP     <FSIZE,FFREE>           ; RESTORE FROM STACK
     16 024026  000137  022762                  JMP     T1SKIP                  ; IF ERROR, EXIT TEST
     17 024032                          2$:
     18 024032                          ENDSUB
        024032                          L10055:
        024032  104403                          TRAP    C$ESUB
     19
```

```
                                ;**
                                ;       TEST 1, SUBTEST 8
                                ;       LARGE BUFFER I/O CONTROLLER MEMORY TEST
                                ;--

024034                          BGNSUB: 8
024034                          T1.8:
024034  104402                          TRAP    C$BSUB
024036                                  PUSH    <FFREE,FSIZE>           ; SAVE FREE MEMORY ADDRESS AND SIZE
024046  013705  002176                  MOV     TSTTAB,R5               ; GET POINTER TO CONTROLLER TABLE
024052  016504  000004                  MOV     C.VEC(R5),R4            ; GET VECTOR OF CONTROLLER
024056  042704  177000                  BIC     #^C<CT.VEC>,R4          ; MASK OTHER BITS
024062  010501                          MOV     R5,R1                   ; GET INTERRUPT SERVICE LINK
024064  062701  000006                  ADD     #C.JSR,R1               ;
024070                                  SETVEC  R4,R1,#PRIO7            ; SET UP INTERRUPT VECTOR
024070  012746  000340                  MOV     #PRIO7,-(SP)
024074  010146                          MOV     R1,-(SP)
024076  010446                          MOV     R4,-(SP)
024100  012746  000003                  MOV     #3,-(SP)
024104  104437                          TRAP    C$SVEC
024106  062706  000010                  ADD     #10,SP
024112  006204                          ASR     R4                      ; INITIALIZE CONTROLLER WITH SMALLEST
024114  006204                          ASR     R4                      ;       POSITION VECTOR FOR RING
024116  004737  017676                  CALL    CNTINT                  ;       BUFFER AND INTERRUPTS ENABLED
024122  001422                          BEQ     1$                      ; IF ERROR, EXIT
024124  004737  012156                  CALL    HCOMM                   ; ALLOCATE SPACE FOR HOST COMM AREA
024130  010402                          MOV     R4,R2                   ; GET HOST COMM ADDRESS
024132  062702  000100                  ADD     #HC.BF1,R2              ; POINT TO BUFFER AREA
024136  012704  036413                  MOV     #T18MSZ,R4              ; SAVE CONTROLLER MEMORY SIZE
                                ;
                                ;       PERFORM I/O AND DATA COMPARISON
                                ;
024142  012737  100000  002336          MOV     #100000,T18PAT          ; INITIALIZE DATA PATTERN
024150  004737  024206                  CALL    T18BLD                  ; PERFORM I/O
024154  103405                          BCS     1$                      ; IF ERROR, EXIT TEST
024156  012737  077777  002336          MOV     #77777,T18PAT           ; INITIALIZE DATA PATTERN
024164  004737  024206                  CALL    T18BLD                  ; PERFORM I/O
024170                          1$:     POP     <FSIZE,FFREE>           ; RESTORE FREE MEMORY ADDRESS AND SIZE
024200                          ENDSUB
024200                          L10056:
024200  104403                          TRAP    C$ESUB

024202  000137  022762                  JMP     T1SKIP                  ; GO TEST NEXT CONTROLLER

                                ;**
                                ;       T18BLD - BUILD THE DATA PACKETS FOR MAINTENANCE I/O OPERATIONS,
                                ;       WRITE THE PACKETS, READ BACK THE DATA AND PERFORM A DATA COMPARISON.
                                ;
                                ;       INPUTS:
                                ;               R2 - DATA BUFFER LOCATION
                                ;               R4 - CONTROLLER MEMORY SIZE
                                ;--

024206  012701  000020          T18BLD: MOV     #16.,R1                 ; INITIALIZE PATTERN COUNT
024212  012700  000031          T18BL1: MOV     #OP.MWR,R0              ; GET MAINTENANCE WRITE COMMAND
024216  013703  002336                  MOV     T18PAT,R3               ; GET OLD PATTERN
```

```
     024222  005703                          TST     R3                      ; CHECK HIGH BIT
     024224  002402                          BLT     5$                      ; IF BIT 15 SET, BRANCH
     024226  000241                          CLC                             ; SHIFT 0 INTO PATTERN
     024230  000401                          BR      6$                      ;
     024232  000261                  5$:     SEC                             ; SHIFT 1 INTO PATTERN
     024234  006103                  6$:     ROL     R3                      ; SHIFT PATTERN 1 BIT LEFT
     024236  010337  002336                  MOV     R3,T18PAT               ; SAVE PATTERN
     024242  012703  000106                  MOV     #HC.BSZ,R3              ; GET SIZE OF BUFFER (WORDS)
     024246                                  PUSH    <R1,R4,R2>              ; SAVE LOOP COUNT,MEMORY SIZE,ETC
     024254  013722  002336          7$:     MOV     T18PAT,(R2)+            ; MOVE DATA TO BUFFER
     024260  005303                          DEC     R3                      ; DECREMENT WORDS REMAINING
     024262  001374                          BNE     7$                      ; LOOP IF MORE WORDS TO LOAD
     024264  012703  000106                  MOV     #HC.BSZ,R3              ; GET SIZE OF BUFFER (WORDS)
     024270  011602                          MOV     (SP),R2                 ; RESTORE BUFFER ADDRESS
     024272  005001                          CLR     R1                      ; INIT CONTROLLER MEMORY OFFSET
     024274  020403                  T18NXT: CMP     R4,R3                   ; CHECK REMAINING MEMORY
     024276  002001                          BGE     8$                      ; IF MORE THAN BUFFER SIZE, SKIP
     024300  010403                          MOV     R4,R3                   ; ELSE, USE REMAINING COUNT
     024302  006303                  8$:     ASL     R3                      ; CONVERT TO BYTES
     024304  004737  017056                  CALL    LOAD                    ; LOAD INTO CONTROLLER
     024310  001477                          BEQ     T18EX1                  ; IF ERROR, EXIT
     024312  006203                          ASR     R3                      ; CONVERT BYTES TO WORDS
     024314  060301                          ADD     R3,R1                   ; INCREASE OFFSET FOR NEXT BUFFER
     024316  160304                          SUB     R3,R4                   ; REDUCE REMAINING BYTE COUNT
     024320  001365                          BNE     T18NXT                  ; GET NEXT BUFFER
                                     ;
                                     ;       READ DATA BACK FROM CONTROLLER MEMORY AND VERIFY IT
                                     ;
     024322  016604  000002                  MOV     2(SP),R4                ; RESTORE CONTROLLER MEMORY SIZE
     024326  012703  000106                  MOV     #HC.BSZ,R3              ; GET SIZE OF BUFFER (WORDS)
     024332  005001                          CLR     R1                      ; INIT CONTROLLER MEMORY OFFSET
     024334  020403                  T18RNX: CMP     R4,R3                   ; CHECK REMAINING MEMORY
     024336  002001                          BGE     9$                      ; IF MORE THAN BUFFER SIZE, SKIP
     024340  010403                          MOV     R4,R3                   ; ELSE, USE REMAINING COUNT
     024342                          9$:     PUSH    R3                      ; SAVE BUFFER SIZE
     024344  006303                          ASL     R3                      ; CONVERT WORD COUNT TO BYTES
     024346  012700  000030                  MOV     #OP.MRD,R0              ; GET MAINTENANCE READ COMMAND
     024352  004737  017056                  CALL    LOAD                    ; READ DATA FROM CONTROLLER
     024356  001453                          BEQ     T18EX2                  ; IF ERROR, GET OUT NOW
     024360  006203                          ASR     R3                      ; CONVERT BYTE COUNT TO WORDS
     024362  016602  000002                  MOV     2(SP),R2                ; RESTORE DATA BUFFER ADDRESS
     024366  023722  002336          10$:    CMP     T18PAT,(R2)+            ; CHECK DATA READ FROM CONTROLLER
     024372  001411                          BEQ     11$                     ; IF SAME, DATA OK
     024374  162702  000002                  SUB     #2.,R2                  ; GO BACK TO BAD WORD
     024400  062601                          ADD     (SP)+,R1                ; POINT TO END OF DATA READ
     024402  160301                          SUB     R3,R1                   ; POINT TO ADDRESS OF FAILING WORD
     024404                                  ERRDF   39.,ERR039              ; PRINT COMPARE ERROR
     024404  104455                          TRAP    C$ERDF
     024406  000047                          .WORD   39
     024410  000000                          .WORD   0
     024412  011544                          .WORD   ERR039
     024414  000435                          BR      T18EX1                  ; EXIT SUBTEST

     024416  005303                  11$:    DEC     R3                      ; DECREMENT WORDS REMAINING
     024420  001362                          BNE     10$                     ; LOOP IF MORE WORDS TO READ

     024422  011603                          MOV     (SP),R3                 ; RESTORE BUFFER SIZE
```

```
    101 024424  016602  000002                  MOV     2(SP),R2                ; RESTORE BUFFER ADDRESS
    102 024430  012722  125125          12$:    MOV     #125125,(R2)+           ; MOVE DATA TO BUFFER
    103 024434  005303                          DEC     R3                      ; DECREMENT WORDS REMAINING
    104 024436  001374                          BNE     12$                     ; LOOP IF MORE WORDS TO LOAD
    105 024440                                  POP     R3                      ; RESTORE BUFFER SIZE
    106 024442  011602                          MOV     (SP),R2                 ; RESTORE BUFFER ADDRESS
    107 024444  006303                          ASL     R3                      ; CONVERT TO BYTES
    108 024446  012700  000031                  MOV     #OP.MWR,R0              ; WRITE BUFFER
    109 024452  004737  017056                  CALL    LOAD                    ;                       OF BAD PATTERN
    110 024456  001414                          BEQ     T18EX1                  ; IF ERROR, EXIT
    111
    112 024460  006203                          ASR     R3                      ; CONVERT TO WORDS
    113 024462  060301                          ADD     R3,R1                   ; INCREASE OFFSET FOR NEXT BUFFER
    114 024464  160304                          SUB     R3,R4                   ; REDUCE REMAINING WORD COUNT
    115 024466  001322                          BNE     T18RNX                  ; GET NEXT BUFFER
    116
    117 024470                                  POP     <R2,R4,R1>              ; RESTORE MEMORY SIZE & PATTERN #
    118 024476  005301                          DEC     R1                      ; CHECK PATTERN NUMBER
    119 024500  001244                          BNE     T18BL1                  ; IF PATTERNS LEFT, DO NEXT PATTERN
    120 024502  000241                          CLC                             ; SUCCESSFUL COMPLETION FLAG
    121 024504  000207                          RETURN                          ; SUCCESS EXIT
    122
    123 024506                          T18EX2: POP     R3                      ; RESTORE STACK
    124 024510                          T18EX1: POP     <R2,R4,R1>              ;                       AFTER ERROR
    125 024516  000261                  T18EX:  SEC                             ; ERROR FLAG
    126 024520  000207                          RETURN                          ;
    127
    128 024522                                  ENDTST
        024522                          L10046:
        024522  104401                          TRAP    C$ETST
    129
```

```
                                .SBTTL  TEST 2: DISK RESIDENT DIAGNOSTIC TEST

                                        BGNTST
024524                          T2::
                                
                                        MOV     #2,R1                   ; INITIALIZE TEST PARAMETERS
                                        CALL    TINIT                   ;
                                        MOV     CTABS,TSTTAB            ; GET ADDRESS TO 1ST CONTROLLER TABLE

                                1$:     CALL    RESET                   ; RESET ALL CONTROLLERS
                                        PUSH    <FFREE,FSIZE>           ;
                                        MOV     #1,R1                   ; RUN DM PROGRAM IN
                                        CALL    RUNDM                   ; ONE CONTROLLER ONLY
                                        BEQ     2$                      ;
                                        CALL    RESPDM                  ;
                                2$:     POP     <FSIZE,FFREE>           ;
                                        ADD     #C.SIZE,TSTTAB          ; MOVE TO NEXT CONTROLLER
                                        TST     @TSTTAB                 ; CHECK IF ANY MORE CONTROLLER TABLES
                                        BNE     1$                      ;

                                        ENDTST
024620                          L10057:
024620  104401                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST 3: DISK FUNCTION TEST
                                        
                                                BGNTST
024622                                  T3::
024622
024622  012701  000003                            MOV     #3,R1                   ; INITIALIZE TEST PARAMETERS
024626  004737  013040                            CALL    TINIT                   ; INITIALIZE CONTROLLERS
024632  013737  002172  002176                    MOV     CTABS,TSTTAB            ; GET ADDRESS OF 1ST TABLE ADDRESS
024640  013701  002174                            MOV     CTRLRS,R1               ; GET COUNT OF CONTROLLERS TO TEST
024644  004737  013104                            CALL    RUNDM                   ; LOAD AND START DM PROGRAM
024650  001402                                    BEQ     1$                      ; EXIT ON ERROR
024652  004737  013200                            CALL    RESPDM                  ; WAIT FOR RESPONSE FROM CONTROLLERS
024656                                  1$:
                                                .EVEN
                                        
                                                ENDTST
024656                                  L10060:
024656
024656  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.
                                ;--
024660                          BGNHRD
024660  000022                  .WORD L10061-L$HARD/2
024662                  L$HARD::

024662                          TABLE                                   ;START A TABLE DEFINITION
024662                  ITEM H.UBA      2       <CSR ADDRESS>
024662                  ITEM H.VEC      2       <CONTROLLER VECTOR>
024662                  ITEM H.BRL      2       <BR LEVEL>
024662                  ITEM H.DRV      2       <DRIVE NUMBER>

        020000          HM.CYL  == BIT13                        ; TEST CUSTOMER DATA AREA
024662                  END
                                                                ; CSR ADDRESS OF CONTROLLER?
024662                          GPRMA   MSGUBA,H.UBA,0,160000,177774,YES
024662  000031                  .WORD   T$CODE
024664  024726                  .WORD   MSGUBA
024666  160000                  .WORD   T$LOLIM
024670  177774                  .WORD   T$HILIM
                                                                ; VECTOR?
024672                          GPRMA   MSGVEC,H.VEC,0,4,774,YES
024672  001031                  .WORD   T$CODE
024674  024760                  .WORD   MSGVEC
024676  000004                  .WORD   T$LOLIM
024700  000774                  .WORD   T$HILIM
                                                                ; BR LEVEL?
024702                          GPRMD   MSGBRL,H.BRL,D,-1,4..7..YES
024702  002052                  .WORD   T$CODE
024704  024767                  .WORD   MSGBRL
024706  177777                  .WORD   -1
024710  000004                  .WORD   T$LOLIM
024712  000007                  .WORD   T$HILIM
                                                                ; DRIVE #?
024714                          GPRMD   MSGLDR,H.DRV,D,-1,0..255..YES
024714  003052                  .WORD   T$CODE
024716  025000                  .WORD   MSGLDR
024720  177777                  .WORD   -1
024722  000000                  .WORD   T$LOLIM
024724  000377                  .WORD   T$HILIM

024726                          ENDHRD
                                .EVEN
024726                  L10061:

024726  103 123 122     MSGUBA: .ASCIZ  \CSR ADDRESS OF CONTROLLER\
024760  126 105 103     MSGVEC: .ASCIZ  \VECTOR\
024767  102 122 040     MSGBRL: .ASCIZ  \BR LEVEL\
025000  104 122 111     MSGLDR: .ASCIZ  \DRIVE #\
```

J16

SEQ 0204

```
CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38  Page 154 1
HARDWARE PARAMETER CODING SECTION


                                        .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.
                                ;--

025010                                  BGNSFT
025010  000003                          .WORD L10062-L$SOFT/2
025012                          L$SOFT::
025012                          TABLE                                   ; START A TABLE DEFINITION
025012                          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
025012                          END

                                                                        ; ENTER MANUAL INTERVENTION MODE
                                                                        ;       IN TEST 2?

025012                                  GPRML S.MAN,SO.BIT,SM.MAN,YES
025012  000130                          .WORD   T$CODE
025014  025020                          .WORD   S.MAN
025016  000200                          .WORD   SM.MAN
025020                                  ENDSFT
                                        .EVEN
025020                          L10062:

025020  105     116     124     S.MAN:  .ASCIZ  \ENTER MANUAL INTERVENTION MODE IN TEST 2\
                                        .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  000060'                         .WORD T$FREE
000042  000006                         .WORD T$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  000004                               .WORD   L10065-./2-1
   000050                               L10063:
15 000050  172150                               .WORD   17215C                  ; CSR ADDRESS
17 000052  000154                               .WORD   154                     ; VECTOR ADDRESS
18 000054  000005                               .WORD   5.                      ; BR LEVEL
20 000056  000000                               .WORD   0.                      ; DRIVE NUMBER
24 000060                                       ENDPTAB
   000060                               L10065:
25 000060                                       ENDSETUP
26         000001                       .END
```

Symbol table

```
ADR    = 000020 G
ALOCM    012132
ARG.CT= 000000
ASS    = 000001
ASSEMB= 000010
BAS      004105
BASLN    004106
BASL1    004024
BASL2    004042
BASL3    004070
BASN0    003724
BASN1    003743
BASN2    003764
BASN3    004004
BELL   = 000007 G
BIT0   = 000001 G
BIT00  = 000001 G
BIT01  = 000002 G
BIT02  = 000004 G
BIT03  = 000010 G
BIT04  = 000020 G
BIT05  = 000040 G
BIT06  = 000100 G
BIT07  = 000200 G
BIT08  = 000400 G
BIT09  = 001000 G
BIT1   = 000002 G
BIT10  = 002000 G
BIT11  = 004000 G
BIT12  = 010000 G
BIT13  = 020000 G
BIT14  = 040000 G
BIT15  = 100000 G
BIT2   = 000004 G
BIT3   = 000010 G
BIT4   = 000020 G
BIT5   = 000040 G
BIT6   = 000100 G
BIT7   = 000200 G
BIT8   = 000400 G
BIT9   = 001000 G
BLDCMD   017176
BLDC0    017244
BLDC1    017252
BNE    = 000400 G
BRLEV    002334
CALR1    015036
CALR2    015064
CALR3    015162
CALR4    015176
CALR5    015252
CALR6    015330
CALR7    015344
CALR8    015362
CALR9    015400
CF.ATN= 000200
CF.MSC= 000100
CF.OTH= 000040
CF.SHC= 000002
CF.THS= 000020
CF.576= 000001
CLOG     020724
CLRBFL   017364
CLRBUF   017336
CNTINT   017676
CNTIST   020272
CNTRSD   002330
CNTRSP   020566
CNTSRV   017554 G
CON.A    012566
CON.A1   012572
CON.A2   012614
CON.D    012616
CON.H    012630
CON.N    012654
CON.N1   012660
CON.O    012642
CON.QU   012546
CON.QX   012564
CON.R    012676
CON.R1   012720
CON.S    012734
CON.S1   012740
CR     = 000015 G
CRLF     002440
CTABER   022606
CTABS    002172 G
CTRLRS   002174
CT.AVL= 100000
CT.BRL= 007000
CT.CMD= 000004
CT.MSG= 000010
CT.REQ= 000020
CT.RN  = 000002
CT.UNT= 000077
CT.VEC= 000777
CT.VER= 000100
C$AU   = 000052
C$AUTO= 000061
C$BRK  = 000022
C$BSEG= 000004
C$BSUB= 000002
C$CEFG= 000045
C$CLCK= 000062
C$CLEA= 000012
C$CLOS= 000035
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000051
C$DRPT= 000024
C$DU   = 000053
C$EDIT= 000003
C$ERDF= 000055
C$ERMR= 000056
C$ERRO= 000060
C$ERSF= 000054
C$ERSO= 000057
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GETB= 000026
C$GETW= 000027
C$GMAN= 000043
C$GPHR= 000042
C$GPLO= 000030
C$GPRI= 000040
C$INIT= 000011
C$INLP= 000020
C$MANI= 000050
C$MEM  = 000031
C$MSG  = 000023
C$OPEN= 000034
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$QIO  = 000377
C$RDBU= 000007
C$REFG= 000047
C$RESE= 000033
C$REVI= 000003
C$RFLA= 000021
C$RPT  = 000025
C$SEFG= 000046
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
C.DR0  = 000016
C.DR1  = 000020
C.DR2  = 000022
C.DR3  = 000024
C.FLG  = 000012
C.HCOM= 000014
C.JAD  = 000010
C.JSR  = 000006
C.REF  = 000032
C.SIZE= 000034
C.TO   = 000026
C.TOH  = 000030
C.UADR= 000000
C.UNIT= 000002
C.VEC  = 000004
DDEF   = 011012
DEBUG  = 000000
DFPTBL   002134 G
DIAG   = 177777
DIAGMC= 000000
DIVIDE   016014
DIV10    016052
DMFRST= 001000
DMMAIN= 000040
DMOVRL= 000004
DMPROG   002200
DMPTR    002340
DMRQA    014564
DMRQB    014626
DMRQD    014720
DMRQE    015032
DMRQ0    013770
DMRQ2    014110
DMRQ2E   014546
DMRQ2M   014124
DMRQ2N   014470
DMRQ2Q   014302
DMRQ2R   014444
DMRQ2V   014370
DMRQ2W   014520
DMRQ2X   014542
DMRQ20   014144
DMRQ22   014216
DMRQ23   014400
DMRQ7    014560
DMTRLN= 000000
DSPSIZ= 000017
DTABS    002170 G
DT.AVL= 100000
DT.UNT= 000077
DUP    = 001000
DU.DFL= 020000
DU.FTL= 050000
DU.INF= 030000
DU.QUE= 010000
DU.SPC= 060000
DU.TER= 040000
D.BE   = 000040
D.CYL  = 000400
D.DC   = 000002
D.DCA  = 000001
D.DCY  = 020000
D.DRV  = 000000
D.ECC  = 010000
D.IW   = 040000
D.RET  = 001000
D.RO   = 004000
D.SEQ  = 000100
D.SIZE= 000004
D.TR   = 000020
D.UNIT= 000002
D.WC   = 000010
D.WCA  = 000004
D.WO   = 002000
D.ZERO= 140200
EF.BBR= 000200
EF.BBU= 000100
EF.CON= 000036 G
EF.LOG= 000040
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.SEX= 000020
EF.STA= 000040 G
ERRBLK   002156 G
ERRC     013006
ERRD     013020
ERRLIM   002643
ERRME1   002477
ERRMSG   002154 G
ERRMSX   014714
ERRNBR   002152 G
ERRTYP   002150 G
ERRVEC= 000004 G
ERR.SZ= 000011
ERR.TB   012756
ERR.TN   011760 G
ERR001   011002 G
ERR002   011020 G
ERR003   011036 G
ERR004   011054 G
ERR006   011066 G
ERR008   011100 G
ERR014   011116 G
ERR021   011134 G
ERR022   011204 G
ERR023   011232 G
ERR024   011312 G
ERR025   011326 G
ERR026   011344 G
ERR027   011364 G
ERR028   011402 G
ERR029   011414 G
ERR030   011430 G
ERR031   011444 G
ERR032   011456 G
ERR033   011474 G
ERR035   011502 G
ERR036   011520 G
ERR038   011532 G
ERR039   011544 G
ERR21A   011172
ERR21E   011202
ERR23A   011252
ERR23B   011272
EVL    = 000004 G
E$END  = 002100
E$LOAD= 000035
FFREE    002160 G
FMEM     002164
FMEMS    002166
FMERR    012120
FRMTT    002435
FSIZE    002162 G
F$AU   = 000015
```

Symbol table

```
F$ALTO= 000020        HC.MPK= 000020        KW.VEC  002262        L$LADP  002026 G      L10045  022722
F$BGN  = 000040        HC.MSG= 000004        KW11I   017572 G      L$LAST  000044RG  003 L10046  024522
F$CLEA= 000007        HC.PSZ= 000060        LDDM    013126        L$LOAD  002100 G      L10047  023102
F$DU   = 000016        HC.RSZ= 000004        LDNEXT  013160        L$LUN   002074 G      L10050  023146
F$END  = 000041        HC.SIZ= 000314        LF     = 000012 G     L$MREV  002050 G      L10051  023264
F$HARD= 000004        HM.CYL= 020000 G      LOAD    017056        L$NAME  002000 G      L10052  023300
F$HW   = 000013        HOE    = 100000 G     LOADB   016744        L$PRIO  002042 G      L10053  023730
F$INIT= 000006        H.BRL  = 000004       LOADDM  016524        L$PROT  021472 G      L10054  023752
F$JMP  = 000050        H.DRV  = 000006       LOADER  017052        L$PRT   002112 G      L10055  024032
F$MOD  = 000000        H.UBA  = 000000       LOADE1  017042        L$REPP  002062 G      L10056  024200
F$MSG  = 000011        H.VEC  = 000002       LOADTX  016740        L$REV   002010 G      L10057  024620
F$PROT= 000021        IBE    = 010000 G     LOADX   017164        L$RPT   021334 G      L10060  024656
F$PWR  = 000017        ICONT  = 000002 G     LOADX1  017154        L$SOFT  025012 G      L10061  024726
F$RPT  = 000012        IDU    = 000040 G     LOE    = 040000 G     L$SPC   002056 G      L10062  025020
F$SEG  = 000003        IER    = 020000 G     LOG    = 000001       L$SPCP  002020 G      L10063  000050R     003
F$SOFT= 000005        IFLAGS  002202 G      LOT    = 000010 G     L$SPTP  002024 G      L10065  000060R     003
F$SRV  = 000010        INITBL  002312        LPNT    012430        L$STA   002030 G      MD.CMP= 040000
F$SUB  = 000002        INITXX  022574        LPNTB   012402        L$SW    002146 G      MD.CWB= 000010
F$SW   = 000014        INIT1   021772        LPNTF   012372        L$TEST  002114 G      MD.ERR= 010000
F$TEST= 000001        INIT2   022154        LPNTS   012422        L$TIML  002014 G      MD.EXP= 100000
GETCDN  015540        INIT3   022212        LPNTX   012412        L$UNIT  002012 G      MD.FEU= 000001
GETCNT  015474        INIT4   022246        LT1L1   016634        L10000  002144        MD.IMF= 000002
GETCNX  015500        INIT6   022560        LT1L2   016662        L10001  002150        MD.NXU= 000001
GETCXX  015546        INTRCV  002214        LT11    016646        L10002  011016        MD.PRI= 000001
GTDRVT  015404        INTSRV  017564 G      L$ACP   002110 G      L10003  011034        MD.RIP= 000001
G$CNTO= 000200        INTSTO  002736        L$APT   002036 G      L10004  011052        MD.SCH= 004000
G$DELM= 000372        INTST1  003042        L$AU    022716 G      L10005  011064        MD.SCL= 002000
G$DISP= 000003        IPADRS  002246        L$AUT   002070 G      L10006  011076        MD.SEC= 000100
G$EXCP= 000400        IREST  = 000004 G     L$AUTO  022674 G      L10007  011114        MD.SEQ= 000020
G$HILI= 000002        ISR    = 000100 G     L$CCP   002106 G      L10010  011132        MD.SER= 000400
G$LOLI= 000001        ISTRT  = 000010 G     L$CLEA  022676 G      L10011  011202        MD.SPD= 000001
G$NO   = 000000        ISTRTH= 000020 G      L$CO    002032 G      L10012  011230        MD.SSH= 000200
G$OFFS= 000400        IXE    = 004000 G     L$DEPO  002011 G      L10013  011310        MD.SWP= 000004
G$OFSI= 000376        I$AU   = 000041       L$DESC  002366 G      L10014  011324        MD.VOL= 000002
G$PRMA= 000001        I$AUTO= 000041        L$DESP  002076 G      L10015  011342        MD.WBN= 000100
G$PRMD= 000002        I$CLN  = 000041       L$DEVP  002060 G      L10016  011362        MD.WBV= 000400
G$PRML= 000000        I$DU   = 000041       L$DISP  002124 G      L10017  011400        MEMFIL  014000
G$RADA= 000140        I$HRD  = 000041       L$DLY   002116 G      L10020  011412        MESSG   003574
G$RADB= 000000        I$INIT= 000041        L$DTP   002040 G      L10021  011426        MLDRER  022622
G$RADD= 000040        I$MOD  = 000041       L$DTYP  002034 G      L10022  011442        MSCP   = 000000
G$RADL= 000120        I$MSG  = 000041       L$DU    022710 G      L10023  011454        MSGBRL  024767
G$RADO= 000020        I$PROT= 000040        L$DUT   002072 G      L10024  011472        MSGLDR  025000
G$XFER= 000004        I$PTAB= 000041        L$DVTY  002342 G      L10025  011500        MSGPKL  011704
G$YES  = 000010        I$PWR  = 000041       L$EF    002052 G      L10026  011516        MSGPKT  011660
HCOMM   012156        I$RPT  = 000041       L$ENVI  002044 G      L10027  011530        MSGUBA  024726
HC.BF1= 000100        I$SEG  = 000041       L$ERRT  002150 G      L10030  011542        MSGVEC  024760
HC.BF2= 000206        I$SETU= 000041        L$ETP   002102 G      L10031  011564        MXFERP  002566
HC.BSZ= 000106        I$SFT  = 000041       L$EXP1  002046 G      L10032  012116        NCON    012104
HC.CCT= 000012        I$SRV  = 000041       L$EXP4  002064 G      L10033  017552        NCONF   012530
HC.CEV= 000014        I$SUB  = 000041       L$EXP5  002066 G      L10034  017562        NCONS   012506
HC.CMD= 000010        I$TST  = 000041       L$HARD  024662 G      L10035  017570        NOCLOC  003647
HC.CPK= 000020        J$JMP  = 000167       L$HIME  002120 G      L10036  017612        NXMAD   002276
HC.ESZ= 000004        KW.BRL  002260        L$HPCP  002016 G      L10037  021470        NXMI    017544 G
HC.INT= 000000        KW.CSR  002256        L$HPTP  002022 G      L10041  022672        OP.ABO= 000001
HC.ISZ= 000004        KW.EL   002266        L$HW    002154 G      L10042  022674        OP.ACC= 000020
HC.MCT= 000006        KW.HZ   002264        L$ICP   002104 G      L10043  022706        OP.AVA= 000100
HC.MEM= 000014        KW.OUT= 000105 G      L$INIT  021500 G      L10044  022714        OP.AVL= 000010
```

D1

Symbol table

```
OP.CCD= 000021
OP.CMP= 000040
OP.DUP= 000101
OP.ELP= 000003
OP.END= 000200
OP.ERS= 000022
OP.ESP= 000002
OP.FLU= 000023
OP.GCS= 000002
OP.GSS= 000001
OP.GUS= 000003
OP.MRD= 000030
OP.MWR= 000031
OP.ONL= 000011
OP.RD = 000041
OP.RLC= 000103
OP.RPL= 000024
OP.RSD= 000005
OP.SCC= 000004
OP.SEX= 000007
OP.SHC= 000102
OP.SSD= 000004
OP.SUC= 000012
OP.WR = 000042
OSTRE   012544
OSTRNG  012476
O$APTS= 000000
O$AU  = 000000
O$BGNR= 000001
O$BGNS= 000001
O$DU  = 000000
O$ERRT= 000001
O$GNSW= 000001
O$POIN= 000001
O$SETU= 000001
PB      012276
PF      012252
PNT   = 001000 G
PNTERR  016360
PNTNUM  015552
PNTNUS  015560
PNTPKL  011616
PNTPKT  011566
PRI   = 002000 G
PRINTC  012216
PRI00 = 000000 G
PRI01 = 000040 G
PRI02 = 000100 G
PRI03 = 000140 G
PRI04 = 000200 G
PRI05 = 000240 G
PRI06 = 000300 G
PRI07 = 000340 G
PS      012346
PTYPE   002306
PY      012322
P.BCNT= 000014

P.BUFF= 000020
P.CHVR= 000023
P.CMST= 000020
P.CNTF= 000016
P.CNTI= 000024
P.CPSP= 000042
P.CRF = 000000
P.CSVR= 000022
P.CTMO= 000020
P.CYLS= 000050
P.DEXT= 000014
P.DFLG= 000017
P.DMDI= 000024
P.DPRG= 000020
P.DTMO= 000024
P.ELGF= 000034
P.FBBK= 000034
P.FLGS= 000011
P.GRPS= 000046
P.HSTI= 000020
P.HTMO= 000020
P.LBN = 000034
P.MEDI= 000034
P.MLUN= 000014
P.MOD = 000012
P.OPCD= 000010
P.OTRF= 000014
P.OVRL= 000034
P.RBN = 000014
P.RBNS= 000056
P.RCTC= 000057
P.RCTS= 000054
P.RGID= 000034
P.RGOF= 000040
P.SHST= 000042
P.SHUN= 000040
P.STS = 000012
P.TIME= 000024
P.TRKS= 000044
P.UADR= 000020
P.UNFL= 000016
P.UNIT= 000004
P.UNSZ= 000044
P.UNTI= 000024
P.USEF= 000022
P.VRSN= 000014
P.VSER= 000050
READDT  013076
RESET   020744
RESPCT  013212
RESPDM  013200
RG.FLG= 040000
RG.OWN= 100000
RNTIM   00??43
RNTIME  021046
RNTIMX  021226
RNTIM1  002466

RNTIM2  002474
RPTMSG  021434
RPTXX   021414
RSPDRP  013410
RSPDSP  013732
RSPERR  013474
RSPIN   013426
RSPMWR  013446
RSPNRP  013374
RSPNTO  013340
RSPNXT  013342
RSPOU   013514
RSPOUT  013630
RSPOU2  013676
RSPOU3  013704
RSPPTW  013506
RSPPT2  013516
RSPPT3  013566
RSPRPT  013372
RSPTM   013276
RSPTMO  013326
RSP.CK  020546
RSP.S1  020500
RSP.S2  020512
RSP.S3  020532
RUNDM   013104
SAMVEC  022656
SA.BST= 000374
SA.CMD= 034000
SA.CME= 000070
SA.CM1= 004000
SA.CNT= 000360
SA.CTP= 003400
SA.DI = 000400
SA.ERC= 003777
SA.ERR= 100000
SA.GO = 000001
SA.INE= 000200
SA.INT= 000200
SA.LFC= 000002
SA.MCV= 000017
SA.MP = 000100
SA.MSE= 000007
SA.MSG= 003400
SA.MS1= 000400
SA.NV = 002000
SA.NVE= 000400
SA.PRG= 000001
SA.QB = 001000
SA.SM = 000040
SA.STE= 000200
SA.STP= 100000
SA.S1 = 004000
SA.S2 = 010000
SA.S3 = 020000
SA.S4 = 040000
SA.TST= 100000

SA.VCE= 000177
SA.VEC= 000177
SA.WRP= 040000
SET00   017626
SET01   017634
SET02   017642
SETTO   017614
SFPTBL  002146 G
SM.IW = 040000 G
SM.LOG= 001000 G
SM.MAN= 000200 G
SM.SSF= 000400 G
SNDCMD  017270
SND.S1  002314
SND.S2  002320
SND.S3  002324
SO.BIT= 000000
SSTEP4  002326
STIME   002272
ST.ABO= 000002
ST.AOL= 000400
ST.AVL= 000004
ST.CMD= 000001
ST.CMP= 000007
ST.CNT= 000012
ST.DAT= 000010
ST.DIA= 000037
ST.DRV= 000013
ST.HST= 000011
ST.MFE= 000005
ST.MSK= 000037
ST.OFL= 000003
ST.SUB= 000040
ST.SUC= 000000
ST.WPR= 000006
SVCGBL= 000000
SVCINS= 000000
SVCSUB= 000000
SVCTAG= 000000
SVCTST= 000000
S$LSYM= 010000
S.MAN   025020
TEMP    002216
TEST1 = ****** GX
TEST2 = ****** GX
TEST3 = ****** GX
TINDEX= 000002
TINIT   013040
TNAMES  013000
TNUM    002204
TOOMER  022640
TSTTAB  002176
TTYOUT  002310
T$ARGC= 000002
T$CODE= 000130
T$ERRN= 000047
T$EXCP= 000000

T$FLAG= 000040
T$FREE= 000060R      003
T$GMAN= 000000
T$HILI= 000377
T$LAST= 000001
T$LOLI= 000000
T$LSYM= 010000
T$LTNO= 000003
T$NEST= 177777
T$NS0 = 000005
T$NS1 = 000002
T$PCNT= 000000
T$PTAB= 010064
T$PTHV= 000001
T$PTNU= 000001
T$SAVL= 177777
T$SEGL= 177777
T$SIZE= 000006
T$SUBN= 000000
T$TAGL= 177777
T$TAGN= 010066
T$TEMP= 000005
T$TEST= 000003
T$TSTM= 177777
T$TSTS= 000001
T$$AU = 010045
T$$AUT= 010042
T$$CLE= 010043
T$$DAT= 010065
T$$DU = 010044
T$$HAR= 010061
T$$HW = 010000
T$$INI= 010041
T$$MSG= 010032
T$$PC = 000001
T$$PRO= 010040
T$$PTA= 01006?
T$$RPT= 010037
T$$SOF= 010062
T$$SRV= 010036
T$$SUB= 010056
T$$SW = 010001
T$$TES= 010060
T1      022724 G
T1NEXT  022742
T1NOW   023002
T1SKIP  022762
T1.1    023006
T1.2    023104
T1.3    023150
T1.4    023266
T1.5    023302
T1.6    023732
T1.7    023754
T1.8    024034
T18BLD  024206
T18BL1  024212
```

```
CZUDHA0 UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38  Page 157 4
Symbol table

T18EX   024516          T2PNTB  016150          UF.WBN= 000100          XPKT1   010301          X26     006331
T18EX1  024510          T2PNTD  016222          UF.WPH= 020000          XPKT2   010606          X27     006530
T18EX2  024506          T2PNTO  016200          UF.WPS= 001000          XSA     010635          X28     006652
T18MSZ= 036413          T2PNTW  016120          UF.576= 000004          X$ALWA= 000000          X29     006727
T18NXT  024274          T2WARN  003057          URNING  002210          X$FALS= 000040          X3      004340
T18PAT  002336          T2WRO   002302          URUN    002206          X$OFFS= 000400          X3A     004117
T18RNX  024334          T2WRR   002300          UTOT1   014576          X$TRUE= 000020          X31     007066
T2      024524 G        T3      024622 G        UTOT1A  014612          X1      004202          X32     007210
T2CMS1  003156          T4OPT7  002432          UTOT2   014616          X1A     004117          X35     007321
T2CMS5  003553          UAM     = 000200 G      WAITMS  017400          X14     004731          X36     007401
T2DR    002304          UCNT    002212          WCHNG   021240          X2      004271          X38     007526
T2GND1  016252          UF.CMR= 000001          WCHNGD  002332          X2A     004117          X39     007732
T2GND2  016254          UF.CMW= 000002          XFCE    010126          X22     005332          X4      004426
T2GND3  016314          UF.INA= 040000          XFRU    010672          X23A    005465          X6      004571
T2GNE   016350          UF.RPL= 100000          XFRUA   010741          X23B    005777          X8      004656
T2GNUM  016232          UF.SCH= 004000          XMSG1   010200          X24     006013          X8A     004117
T2GNX   016344          UF.SCL= 002000          XMSG2   010234          X25     006200          $PATCH  000000RG-    003
T2PNT   016160

. ABS.  025072    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
DM      000000    002   (RW,I,LCL,ABS,CON)
END     000060    003   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 661
Work  file writes: 595
Size of work file: 28754 Words  ( 113 Pages)
Size of core pool: 14080 Words  ( 55 Pages)
Operating  system: RT-11 (Under RTEM-11)

Elapsed time: 00:04:23.00
ZUDHA0,ZUDHA0/C=SVC34R.MLB,ZUDHA0.MEM,CZUDH0.MAC
```

CZUDHAO UDA50A/KDA50-Q BASIC SU MACRO V05.01b Sunday 07-Oct-84 10:38 Page S 1
Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
BIT3     88-80    94-24    95-26    96-24
BIT4     88-80    94 21    95-25    96 25
BIT5     88-80    95-24
BIT6     88-80    94-19    95-23
BIT7     88 80    95-31   111-56
BIT8     88-80    95-22
BIT9     88-80    95-21
BLDC0   122-26   122-280
BLDC1   122-300  122-32
BLDCMD  102-116  102-134  120-66   121-18   122-170
BOE      88-80
BRLEV    96-1010 148-16*  148-74*  148-75
C$AU     84-1530 142-14
C$AUTO   84-1530 139-12
C$BRK    84-1530 102-47   125-27   129-17   130 30   134-14   148-36
C$BSEG   84-1530
C$BSUB   84-1530 144-6    145-8    146-7    147-8    148-7    149-9    150-7    151-7
C$CEFG   84-1530
C$CLCK   84-1530 137-127  137-129
C$CLEA   84-1530 140-16
C$CLOS   84-1530
C$CLP1   84-1530 144-16
C$CVEC   84-1530 129-33   144-12   148-80   148-85
C$DCLN   84-1530  99-11   137-105  138-11   138-21   138-30   138-39
C$DODU   84-1530
C$DRPT   84-1530 102-59   119-26
C$DU     84-1530 141-12
C$EDIT   84-1530  84-201
C$ERDF   84-1530 102-26   102-44   102-94   102-112  112-25   120-91   121-35   125-35   125 44   128-76   128-96   128-122  129-36
         129-93  130-40   130-51   134-22   144-15   146-29   148-67   148-77   151 94
C$ERHR   84-1530
C$ERRO   84-1530 119-41
C$ERSF   84-1530  99-9    128-128  128-134  137-103  138-9    138-19   138-28   138-37
C$ERSO   84-1530
C$ESCA   84-1530
C$ESEG   84-1530
C$ESUB   84-1530 144-19   145-17   146-34   147-12   148-86   149-15   150-18   151-32
C$ETST   84-1530 151-128  152-23   153-17
C$EXIT   84-1530 137-353  140-14   143-26
C$GETB   84-1530
C$GETW   84-1530
C$GMAN   84-1530 105-66
C$GPHR   84-1530 137-93   137-184
C$GPLO   84-1530
C$GPRI   84-1530
C$INIT   84-1530 138-42
C$INLP   84-1530
C$MANI   84-1530 105-36
C$MEM    84-1530 137-146
C$MSG    84-1536  98-12   98-17    98-21    98-25    98-29    98-34    98-39    98-53    98-58    98-73    98-77    98-81    98-85
         98-89    98-93    98-97    98 102   98-106   98-111   98-115   98-120   98 124   98 128   98 133   98 186
C$OPEN   84-1530
C$PNTB   84-1530  99-80
C$PNTF   84-1530  99-78
C$PNTS   84-1530  99-84
C$PNTY   84-1530  99-82
```

Cross reference table (CREF V05.01)

```
C$QIO    84 153#
C$RDBU   84-153#
C$REFG   84-153# 137-46   137-52   137-58   137-65
C$RESE   84-153#  84-153#
C$REVI   84-153#  84-201
C$RFLA   84 153# 108-40
C$RPT    84-153# 135-107
C$SEFG   84-153#
C$SPRI   84-153# 137-351  148-20   148-48   148-61   148-70
C$SVEC   84-153# 120-20   129-30   132-14   137-140  144-8    148-19   151-14
C$TPRI   84-153#
C.DR0    94-37#  107-19   112-16   137-77   137-267
C.DR1    94-38#
C.DR2    94-39#
C.DR3    94-40#
C.FLG    94-35#  101-24*  102-13   102-16   102-18   102-70*  102-79   102-99   102-130*  102-131  102-137*  102-140*  108-51*  109-19*
        120-83*  120-84*  121-28*  123-19*  125-18   125-38*  137-256
C.HCOM   94-36#   98-156    99-54*  102-12   122-18   123-13
C.JAD    94-34#
C.JSR    94-33#  120-19   151-13
C.REF    94-43#  102-92   123-14*  123-15
C.SIZE   94-45#  101-32   102-63   137-84   137-99   137-220  137-236  137-256  143-23   152-19
C.TO     94-41#  102-42   102-145  125-15   125-33   130-25   130-36   134-10   134-20   148-32   148-43
C.TOH    94-42#  102-39   125-30   130-33   134-17   148-40
C.UADR   94-30#   98-173  129-28   137-190  146-9    148-9    148-18
C.UNIT   94-31#  101-25   101-26   102-15   119-17   128-121* 137-75*  137-110*  143-19   143-20
C.VEC    94-32#  120-16   137-198  137-202  148-10   148-78   148-83   151-10
CALR1    99-254  111-9#   111-57
CALR2    99-255  111-24#
CALR3    99-256  111-56#
CALR4    99-257  111-68#
CALR5    99-258  111-81#
CALR6    99-259  111-93#
CALR7    99-260  111-105#
CALR8    99-261  111-117#
CALR9    99-262  111-130#
CF.576   91-94#
CF.ATN   91-89#
CF.MSC   91-90#
CF.OTH   91-91#
CF.SHD   91-93#
CF.THS   91-92#
CLOG    128-35   128-43   131-10#
CLRBFL  124-23#  124-25
CLRBUF  102-118  102-136  124-17#
CNTINT  120-23   128-30#  147-10   149-11   151-17
CNTIST  128-67   129-16#
CNTRSD   96-98#   98-56*   98-57   129-44*  129-55*  129-56   130-22*  130-28   145-10*  145-14*  146-11*  148-26*
CNTRSP  128-81   129-51   130-21#  145-11   145 15   146-12   148-27
CNTSRV  126-32#  137-254
CON.A    99-181#  99-297
CON.A1   99-182#  99-184  111-39
CON.A2   99-186   99-188#
CON.D    99-194#  99-298
CON.H    99-202#  99-299
CON.N    99-218#  99-301
```

```
CON.N1   99-219#  99-221
CON.O    99-210#  99-300
CON.QU   99-169#  99-173   99-296
CON.QX   99-171   99-174#
CON.R    99-229#  99-302
CON.R1   99-231   99-236#
CON.S    99-244#  99-303
CON.S1   99-245#  99-247
CR       88-9     98-175   99-71    99-219   105-62   111-32   111-40   111-70   111-83   135-15
CRLF     97-36#   99-73
CT.AVL   94-18#  101-28   137-75   137-110  143-22
CT.BRL   94 16#  148-13
CT.CMD   94-26#  102-18   120-83   121-28   123-19
CT.MSG   94-24#  102-16   102-70   102-130  108-51   109-19   125-18   125-38   126-33
CT.REQ   94-21#  102-79   102-99   102-131  102-137  102-140  120-83
CT.RN    94-27#  102-13   102-70   120-84
CT.UNT   94-14#
CT.VEC   94-15#  120-17   137-203  148-12   148-79   148-84   151-11
CT.VER   94-19#
CTABER  137-200  138-8#
CTABS    96-15#  137-74   137-96   137-168* 137-169* 137-186  143-17   152-10   153-9
CTRLRS   96-16#  137-170* 137-260* 153-10
D.BE     95-24#
D.CYL    95-22#
D.DC     95-28#   95-30
D.DCA    95-29#
D.DCY    95-17#
D.DRV    95-34#
D.ECC    95-18#   95-30
D.IW     95 16#   95-31
D.RET    95-21#   95-30
D.RO     95-19#
D.SEQ    95-23#
D.SIZE   95-78#  137-158  137-296
D.TR     95-25#
D.UNIT   95-35#  105-47   107-23   109-24   109-26   112-31   119-23   137-81*  137-111*
D.WC     95-26#   95-30
D.WCA    95-27#
D.WO     95-20#
D.ZERO   95-31#
DDEF     95-30#
DEBUG     2-27#   87-17    99-304   102-32   120-71   125-22   143-13   152-6    153-5    155-19   155-50   155-68
DFPTBL   86-12#
DIAG     90-16#  122-27
DIAGMC   84-153   84-153
DIV10   116-20#
DIVIDE  114-40   114-53   115-17#  133-20   133-22   133-24
DMFRST   95-87#
DMMAIN   95-86#  120-34   120-36   120-60
DMOVRL   95-85#
DMPROG   96-18#   98-179   98-180  100-24*  109-31   109-32   111-12   111-13   120-33   120-35   120-64   120-67   120-69   120-70
DMPTR    96-105# 120-50*  120-54
DMRQ0   102-151  104-29#
DMRQ2   102-154  105-32#
DMRQ20  105 37   105-40#
DMRQ22  105-42   105-52#
```

```
DMRQ23  105-72   105-84#
DMRQ2E  105-79   105-88   105-91   105-93   105-108  105-113# 118-49
DMRQ2M  105-34   105-36#
DMRQ2N  105-98   105-100#
DMRQ2Q  105-53   105-66#  105-114
DMRQ2R  105-86   105-95#
DMRQ2V  105-70   105-78#
DMRQ2W  105-101  105-105  105-107#
DMRQ2X  105-35   105-39   105-80   105-111#
DMRQ7   102-159  106-10#
DMRQA   102-162  107-18#
DMRQB   102-163  102-164  108-26#
DMRQD   102-165  109-19#
DMRQE   102-166  110-14#
DMTRLN   95-84#
DSPSIZ  102-110  102-183#
DT.AVL   95-15#  107-25   137-81   137-111
DT.UNT   95-13#
DTABS    96-14#  137-154* 137-155* 137-266  137-296* 137-297*
DU.DFL   93-101#
DU.FTL   93-104#
DU.INF   93-102#
DU.QUE   93-100#
DU.SPC   93-105# 102-106
DU.TER   93-103#
DUP      90-17#  122-22
E$END    84-153#
E$LOAD   84-153#  84-201
EF.BBR   91-80#
EF.BBU   91-81#
EF.CON   88-8#   137-58
EF.LOG   91-82#
EF.NEW   88-8#
EF.PWR   88-8#   137-65
EF.RES   88-8#   137-52
EF.SEX   91-83#
EF.STA   88-8#   137-46
ERR.SZ   99-230   99-264#
ERR.TB   99-233   99-254#  99-264
ERR.TN   98-165# 119-40
ERR001   98-10#  138-9
ERR002   98-15#  138-19
ERR003   98-19#  138-28
ERR004   98-23#   99-9
ERR006   98-27#  137-103
ERR008   98-32#  138-37
ERR014   98-37#  128-122  128-128  128-134
ERR021   98-42#  130-51
ERR022   98-55#  130-40
ERR023   98-60#  128-96
ERR024   98-75#  128-76
ERR025   98-79#  129-93
ERR026   98-83#  146-29
ERR027   98-87#  134-22
ERR028   98-91#  148-67
ERR029   98-95#  148-77
```

Cross reference table (CREF V05.01)

```
ERR030  98-100# 102-26  125-44
ERR031  98-104# 102-44
ERR032  98-108# 102-112
ERR033  98-113# 102-94  120-91  121-35
ERR035  98 117# 112-25
ERR036  98-122# 125-35
ERR038  98-126# 129-36  144-15
ERR039  98-131# 151-94
ERR21A  98-47   98-50#
ERR21E  98-49   98-52#
ERR23A  98-63#  98-70
ERR23B  98-64   98-68#
ERRBLK  96-8#
ERRC    99-151  99-158  99-280#
ERRD    99-160  99-296#
ERRLIM  97-42#
ERRME1  97-40#  99-156  99-236
ERRMSG  96-8#
ERRMSX 108-49  108-57  108-83#
ERRNBR  96-8#
ERRTYP  96-8#  119-31
ERRVEC  88-12  129-30  129-33  132-14  144-8   144-12
EVL     88-8#
F$AU    84-153# 142-9   142-14
F$AUTO  84-153# 139-10  139-12
F$BGN   84-153#  98-10   98-15   98-19   98-23   98-27   98-32   98-37   98-42   98-55   98-60   98-75   98-79   98-83
        98-87    98-91   98-95   98-100  98-104  98-108  98-113  98-117  98-122  98-126  98-131  98-165  126-11  126-32
       126-42   126-50  135-10  136-9   137-44  137-353 139-10  140-8   140-14  141-8   142-9   143-10  143-26  144-6
       144-6    144-19  145-8   145-8   145-17  146-7   146-7   146-34  147-8   147-8   147-12  148-7   148-7   148-86
       149-9    149-9   149-15  150-7   150-7   150-18  151-7   151-7   151-32  151-128 152-3   152-23  153-3   153-17
       154-11   155-12  157-13  157-14  157-14  157-24  157-25
F$CLEA  84-153# 140-8   140-16
F$DU    84-153# 141-8   141-12
F$END   84-153   84-153  84-153  84-153  84-153  84-153  84-153  84-153  84-153  84-153  84-153  84-153  84-153  84-153
        84-153   84-153  84-153# 98-12   98-17   98-21   98-25   98-29   98-34   98-39   98-53   98-58   98-73   98-77
        98-81    98-85   98-89   98-93   98-97   98-102  98-106  98-111  98-115  98-120  98-124  98 128  98-133  98-186
       126-13   126-35  126-44  126-54  135-93  135-107 137-353 138-42  139-12  140-14  140-16  141-10  141-12  142-12
       142-14   143-10  143-10  143-10  143-26  144-6   144-6   144-19  144-19  145-8   145-8   145-17  145-17  146-7
       146-7    146-34  146-34  147-8   147-8   147-12  147-12  148-7   148-7   148-86  148-86  149-9   149-9   149-15
       149-15   150-7   150-7   150-18  150-18  151-7   151-7   151-32  151-32  151-128 151-128 152-3   152-3   152-3
       152-23   152-23  153-3   153-3   153-3   153-17  153-17  154-41  155-55  157-13  157-14  157-24  157-25
F$HARD  84-153# 154-11  154-41
F$HW    84-153#  86-12   86-23
F$INIT  84-153# 137-44  138-42
F$JMP   84-153# 135-93  135-93  137-353 140-14  141-10  141-10  142-12  142-12  143-26
F$MOD   84-153#
F$MSG   84-153#  98-10   98-12   98-15   98-17   98-19   98-21   98-23   98-25   98-27   98-29   98-32   98-34   98-37
        98-39    98-42   98-53   98-55   98-58   98-60   98-73   98-75   98-77   98-79   98-81   98-83   98-85   98-87
        98-89    98-91   98-93   98-95   98-97   98-100  98-102  98-104  98-106  98-108  98-111  98-113  98-115  98-117
        98-120   98-122  98-124  98-126  98-128  98-131  98-133  98-165  98-186
F$PROT  84-153# 136-9   136-15
F$PWR   84-153#
F$RPT   84-153# 135-10  135-107
F$SEG   84-153#
F$SOFT  84-153# 155-12  155-55
F$SRV   84-153# 126-11  126-13  126-32  126-35  126-42  126-44  126-50  126-54
```

Cross reference table (CREF V05.01)

```
F$SUB    84-153#  144-6    144-19   145-8    145-17   146-7    146-34   147-8    147-12   148-7    148-86   149-9    149-15   150-7
        150-18    151-7    151-32
F$SW     84-153#   87-11    87-21
F$TEST   84-153#  143-10   151-128  152-3    152 23   153-3    153-17
FFREE    96 10#    98-61    99-30    99-34*   100-21*  104-30   104-42   104-46   120-43   128-58   128-88   129-21   137-146* 137-147
        137 154   137-168  137-344  150-8    150-13*  150-15*  151-8    151-31*  152-13   152-18*
FMEM     96-12#   100-21   137-344*
FMEMS    96-13#   100-22   137-345*
FMERR    99-9#     99-32   128-52
FRMTT    97-35#    99-70
FSIZE    96-11#    99-31*  100-22*  104-31   104-44   128-50   137-147* 137-345  150-8    150-13*  150-15*  151-8    151-31*  152-13
        152-18*
G$CNTO   84-153#
G$DELM   84-153#
G$DISP   84-153#
G$EXCP   84-153#
G$HILI   84-153#
G$LOLI   84-153#
G$NO     84-153#  105-66
G$OFFS   84-153#  105-66   154-27   154-30   154-32   154-35   155-32
G$OFSI   84-153#  105-66   154-27   154-30   154-32   154-35   155-32
G$PRMA   84-153#  154-27   154-30
G$PRMD   84-153#  105-66   154-32   154-35
G$PRML   84-153#  155-32
G$RADA   84-153#  105-66
G$RADB   84-153#
G$RADD   84-153#  154-32   154-35
G$RADL   84-153#  155-32
G$RADO   84-153#  154-27   154-30
G$XFER   84-153#
G$YES    84-153#  154-27   154-30   154-32   154-35   155-32
GETCDN  113-17    113-19   113-29#
GETCNT   99-181    99-218   99-229   99-244   113-14#  114-16
GETCNX  113-16#   113-28
GETCXX  113-30    113-32#
GTDRVT  105-43    109-22   112-14#  119-21   137-108
H.BRL   137-194   137-243  154-17#  154-32   154-32   154-32
H.DRV   137-107   137-271  137-281  154-19#  154-35   154-35   154-35
H.UBA   137-191   154-14#  154-27   154-27   154-27
H.VEC   137-197   137-204  137-246  154-16#  154-30   154-30   154-30
HC.BF1   90-68#    90-69   102-117  151-21
HC.BF2   90-69#    90-71    98-157  102-104  102-108* 102-109  102-135
HC.BSZ   90-52#    90-69    90-71   102-120  120-39   120-44   124-20   124-22   151-55   151-60   151-77
HC.CCT   90-60#   123-17*
HC.CEV   90-65#   122-20
HC.CMD   90-59#    90-60    90-62    99-59*
HC.CPK   90-66#    90-68    98-143   99-58    120-67*  120-68*  120-69*  120-70*  121-19*  121-20*  121-21*  121-22*  122 33*  123 15*
        124-19*  124-20*
HC.ESZ   90-50#    90-63
HC.INT   90-54#    90-56
HC.ISZ   90-48#    90-56   128-46
HC.MCT   90-57#   123-16*
HC.MEV   90-62#    90-63    90-65
HC.MPK   90-63#    90-66    98-145   99-56    99-58    102-82   102-87   102-92   120-81   121-26
HC.MSG   90-56#    90-57    90-59    99-57*   129-22
HC.PSZ   90-51#    90-68   122-21   122-29
```

Cross reference table (CREF V05.01)

```
HC.RSZ  90-49#   90-59    90-62
HC.SIZ  90-71#   99-51
HCOMM   99-51#  120-25   151-19
HM.CYL 154-24#
HOE     88-8#
I$AU    84 153# 142-9#   142-14#
I$AUTO  84-153# 139-10#  139-12#
I$CLN   84-153# 140-8#   140-14   140-16#
I$DU    84-153# 141-8#   141-12#
I$HRD  154-11#  154-41#
I$INIT  84-153# 137-44#  137-353  138-42#
I$MOD   84-153#
I$MSG   84-153#  98-10#   98-12#   98-15#   98-17#   98-19#   98-21#   98-23#   98-25#   98-27#   98-29#   98-32#   98-34#   98-37#
        98-39#   98-42#   98-53#   98-55#   98-58#   98-60#   98-73#   98-75#   98-77#   98-79#   98-81#   98-83#   98-85#   98-87#
        98-89#   98-91#   98-93#   98-95#   98-97#   98-100#  98-102#  98-104#  98-106#  98-108#  98-111#  98-113#  98-115#  98-117#
        98-120#  98-122#  98-124#  98-126#  98-128#  98-131#  98-133#  98-165#  98-186#
I$PROT  84-153# 136-9#
I$PTAB  84-153# 157-14   157-14#  157-24   157-24#
I$PWR   84-153#
I$RPT   84-153# 135-10#  135-107#
I$SEG   84-153# 143-10   144-6    145-8    146-7    147-8    148-7    149-9    150-7    151-7    152-3    153-3
I$SETU  84-153# 157-13   157-13#  157-14   157-25   157-25#
I$SFT  155-12# 155-55#
I$SRV   84-153# 126-11#  126-13#  126-32#  126-35#  126-42#  126-44#  126-50#  126-54#
I$SUB   84-153# 143-10   144-6    144-6#   144-19   144-19#  144-19#  145-8    145-8#   145-17   145-17#  145-17#  146-7    146-7#
       146-34   146-34#  146-34#  147-8    147-8#   147-12   147-12#  147-12#  148-7    148-7#   148-86   148-86#  148-86#  149-9
       149-9#   149-15   149-15#  149-15#  150-7    150-7#   150-18   150-18#  150-18#  151-7    151-7#   151-32   151-32#  151-32#
       152-3    153-3
I$TST   84-153# 143-10   143-10#  143-26   144-6    145-8    146-7    147-8    148-7    149-9    150-7    151-7    151-128  151-128#
       151-128# 152-3    152-3#   152-23   152-23#  152-23#  153-3    153-3#   153-17   153-17#  153-17#
IBE     88-8#
ICONT   96-22#  137-62
IDU     88-8#   108-41
IER     88-8#
IFLAGS  96-20#  137-49*  137-55*  137-61*  137-62*
INIT1  137-50   137-56   137-125#
INIT2  137-154#
INIT3  137-168#
INIT4  137-181#
INIT6  137-344#
INITBL  96-89#  129-45
INITXX 137-118  137-351#
INTRCV  96-31#  126-43*  148-24*  148-34   148-47*  148-63
INTSRV 126-42#  148-19
INTSTO  97-52#  148-17
INTST1  97-53#  148-82
IPADRS  96-35#  132-16   137-171  137-227  137-228
IREST   96-23#  137-55
ISR     88-8#
ISTRT   96-24#  137-49
ISTRTH  96-25#  137-61
IXE     88-8#
J$JMP   84-153# 135-93   141-10   142-12
KW.BRL  96-66#  137-136*
KW.CSR  96-65#  102-37   102-51   125-28   126-53*  130-31   132-22   132-24*  133-14   134-15   137-131* 137-135* 137-141* 148-38
KW.EL   96-69#  102-39   102-42   102-53   102-56   125-30   125-33   126-51*  126-52*  127 29   127-30   127-31   130-33   130-36
```

Cross reference table (CREF V05.01)

```
        133-17   133-18   134-17   134-20   137-125* 137-126* 148-40   148-43
KW.HZ    96-68#  127-16   133-19   137-138*
KW.OUT   88-13   126-53   132-24   137-141
KW.VEC   96-67#  137-137* 137-140
KW11I   126-50#  137-140
L$ACP    84-201#
L$APT    84-201#
L$AU    142-9#
L$AUT    84-201#
L$AUTO   84-201  139-10#
L$CCP    84-201#
L$CLEA   84-201  140-8#
L$CO     84-201#
L$DEPO   84-201#
L$DESC   84-201   97-19#
L$DESP   84-201#
L$DEVP   84-201#
L$DISP   84-201   85-10#
L$DLY    84-201#
L$DTP    84-201#
L$DTYP   84-201#
L$DU    141-8#
L$DUT    84-201#
L$DVTY   84-201   97-13#
L$EF     84-201#
L$ENVI   84-201#
L$ERRT   84-201   96-8#
L$ETP    84-201#
L$EXP1   84-201#
L$EXP4   84-201#
L$EXP5   84-201#
L$HARD   84-201  154-11   154-11#
L$HIME   84-201# 104-55
L$HPCP   84-201#
L$HPTP   84-201#
L$HW     84-201   86-12    86-12#
L$ICP    84-201#
L$INIT   84-201  137-44#
L$LADP   84-201#
L$LAST   84-201  156-17#  157-25
L$LOAD   84-201#
L$LUN    84-201# 101-25*  102-15*  112-31*  119-17*  143-19*
L$MREV   84-201#
L$NAME   84-201#
L$PRIO   84-201#
L$PROT   84-201  136-9#
L$PRT    84-201#
L$REPP   84-201#
L$REV    84-201#
L$RPT    84-201  135-10#
L$SOFT   84-201  155-12   155-12#
L$SPC    84-201#
L$SPCP   84-201#
L$SPTP   84-201#
L$STA    84-201#
L$SW     84-201   87-11    87-11#
```

Cross reference table (CREF V05.01)

```
L$TEST  84 2010
L$TIML  84 2010
L$UNIT  84-2010 137-113  137-156  137 299
L10000  86-12    86-230
L10001  87 11    87-210
L10002  98-120
L10003  98-170
L10004  98-210
L10005  98-250
L10006  98-290
L10007  98-340
L10010  98-390
L10011  98-530
L10012  98-580
L10013  98-730
L10014  98-770
L10015  98-810
L10016  98-850
L10017  98-890
L10020  98-930
L10021  98-970
L10022  98-1020
L10023  98-1060
L10024  98-1110
L10025  98-1150
L10026  98-1200
L10027  98-1240
L10030  98-1280
L10031  98-1330
L10032  98-1860
L10033 126-130
L10034 126-350
L10035 126-440
L10036 126-540
L10037 135-93    135-1070
L10041 137-353   138-420
L10042 139-120
L10043 140-14    140-160
L10044 141-10    141-120
L10045 142-12    142-140
L10046 143-26    151-1280
L10047 144-190
L10050 145-170
L10051 146-340
L10052 147-120
L10053 148-860
L10054 149-150
L10055 150-180
L10056 151-320
L10057 152-230
L10060 153-170
L10061 154-11    154-410
L10062 155-12    155 550
L10063 157-140
L10065 157-14    157-240
LDDM    101 230  101 34
```

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
NCONS    99-152#  99-155
NOCLOC   97-67#  137-132
NODOCU    2-39
NXMAD    96-72#  126-12*  129-29*  129-34   132-12*  144-7*   144-13
NXMI    126 11#  129-30   132-14   144 8
O$APTS   84-153#  84-201
O$AU     84-153#  84-201
O$BGNR   84-153#  84-192#  84-201
O$BGNS   84-153#  84-192#  84-201
O$DU     84-153#  84-201
O$ERRT   84-153#  84-192#  84-201
O$GNSW   84 153#  84-192#  84-201
O$POIN   84-153#  84-192   84-192#  84-192#  84-192#  84-192#  84-192#  84-201
O$SETU   84-153#  84-192#  84-201  156-17
OP.ABO   91-20#
OP.ACC   91-21#
OP.AVA   91-39#
OP.AVL   91-22#
OP.CCD   91-23#
OP.CMP   91-24#
OP.DUP   91-40#
OP.ELP   91-46#
OP.END   91-37#  102-78   102-81
OP.ERS   91-25#
OP.ESP   91-45#  120-65
OP.FLU   91-26#
OP.GCS   91-27#
OP.GSS   91-44#
OP.GUS   91-28#
OP.MRD   91-35#  122-25   151-84
OP.MWR   91-36#  120-51   122-23   151-46   151-108
OP.ONL   91-29#
OP.RD    91-30#
OP.RLC   91-42#
OP.RPL   91-31#
OP.RSD   91-48#  102-81   102-133
OP.SCC   91-32#
OP.SEX   91-38#
OP.SHC   91-41#
OP.SSD   91-47#  102-78   102-115
OP.SUC   91-33#
OP.WR    91-34#
OSTRE    99-150   99-157   99-162#
OSTRNG   98-185   99-109   99-149#  99-161  109-36   111-14   111-95   111-107  111-119
P.BCNT   92-11#   93-12#   120-68*  121-20*  124-20*
P.BUFF   92-12#
P.CHVR   93-63#
P.CMST   93-20#
P.CNTF   92-43#   93-60#
P.CNTI   93-64#
P.CPSP   92-30#
P.CRF    92-7#    93-7#    102-92   123-15*
P.CSVR   93-62#
P.CTMO   93-61#
P.CYLS   93-35#
P.DEXT   93-70#
```

Cross reference table (CREF V05.01)

```
P.DFLG  93-71#
P.DMDT  92-59#
P.DPRG  93-72#
P.DTMO  93-73#
P.ELGF  92-28#
P.FBBK  93 13#
P.FLGS  93-10#
P.GRPS  93-34#
P.HSTI  92-27#   93-28#   93-47#
P.HTMO  92-44#
P.LBN   92-14#
P.MEDI  93-30#   93-49#
P.MLUN  93-26#   93-45#
P.MOD   92-10#
P.OPCD  92-9#    93-9#    102-82   122-33*
P.OTRF  92-20#   93-19#
P.OVRL  92-60#  120-69*  120-70*
P.RBN   92-36#
P.RBNS  93-37#
P.RCTC  93-38#
P.RCTS  93-36#
P.RGID  92-52#  121-22*
P.RGOF  92-53#  121-21*
P.SHST  93-32#   93-51#
P.SHUN  92-29#   93-31#   93-50#
P.STS   93-11#  102-87   120-81   121-26
P.TIME  92-46#
P.TRKS  93-33#
P.UADR  92-13#  120-67*  121-19*  124-19*
P.UNFL  92-26#   93-27#   93-46#
P.UNIT  92-8#    93-8#
P.UNSZ  93-52#
P.UNTI  93-29#   93-48#
P.USEF  92-45#
P.VRSN  92-42#   93-59#
P.VSER  93-53#
PB      99-80#   99-96
PF      96-80    99-78#   99-93
PNT     88-8#
PNTERR 108-56   119-14#
PNTNUM  99-195   99-203   99-211   114-15#
PNTNUS 111-31   114-17#
PNTPKL  98-148#  98-152
PNTPKT  98-114   98-141#
PRI     88-8#
PRIC0   88-8#   137-351  148-19   148-20   148-70
PRIC1   88-8#
PRIO2   88-8#
PRIO3   88-8#
PRIO4   88-8#
PRIO5   88-8#
PRIO6   88-8#
PRIO7   84-201   88-8#   120-20   129-30   132-14   137-140  144-8    148-48   151-14
PRINTC  98-175   99-68#   99-172   99-182   99-219   99 245   105-54   105-59   105-62   111-32   111-40   111-70   111-83   114-59
       114-67   117-8    117-22   117-42   133-28   133-33   133-36   135-15
PS      99-84#   99-102
```

```
PTYPE    96-80#   98-184*  99-75    99-93*   99-96*   99-99*   99-102* 111-71*  111-84*
PX       98-184   99-82#   99-99    111-71   111-84
READDT  100-24    100-27#
RESET   100-17    102-12#  137-145  140-11   143-28   152-12
RESPCT  102-12#   102-65
RESPDM  102 10#   102-66   150-12   152-17   153-13
RG.FLG   90-8#    123-16
RG.OWN   90-7#    123-16   123-17
RNTIM    97-37#   133-25
RNTIM1   97-38#   133-29
RNTIM2   97-39#   133-34
RNTIME   98-174   109-27   111-69   111-82   133-14#  135-14
RNTIMX  133-15    133-36#
RPTMSG  135-13    135-95#
RPTXX   135-91#
RSP.CK  129-70    129-79   129-90#
RSP.S1   96-89    129-68#
RSP.S2   96 91    129-75#
RSP.S3   96-93    129-84#
RSPDRP  102-27    102-45   102-70#  102-95   102-113  102-125
RSPDSP  102-124   102-151# 102-183
RSPERR  102-83    102-88   102-94#
RSPIN   102-17    102-78#
RSPMSR  102-80    102-82#
RSPNRP  102-52    102-55   102-57   102-63#
RSPNTO  102-38    102-41   102-43   102-46#
RSPNXT  102-14    102-51#  102-72   102-147
RSPOU   102-19    102-100#
RSPOU2  102-132   102-140#
RSPOU3  102-138   102-141#
RSPOUT  102-100   102-130#
RSPPT2  102-104#
RSPPT3  102-111   102-115#
RSPPTW  102-93    102-99#
RSPRPT  102-54    102-58#
RSPTM   102-25    102-31#
RSPTMO  102-40    102-44#
RUNDM   101-16#   150-10   152-15   153-11
S$LSYM   84-153#   86-23#   87-21#   98-12#   98-17#   98-21#   98-25#   98-29#   98-34#   98-39#   98-53#   98-58#   98-73#   98-77#
         98-81#   98-85#   98-89#   98-93#   98-97#   98-102#  98-106#  98-111#  98-115#  98-120#  98-124#  98-128#  98-133#  98-186#
        105-66   105-66   105-66   105-66#  126-13#  126-35#  126-44#  126-54#  135-107# 138-42#  139-12#  140-16#  141-12#  142 14#
        144-19#  145-17#  146-34#  147-12#  148-86#  149-15#  150-18#  151-32#  151-128# 152-23#  153-17#  154-41#  155-55#
S.MAN   155-32   155-58#
SA.BST   89-86#
SA.CM1   89-31#  149-10
SA.CMD   89-26#
SA.CME   89-56#
SA.CNT   89-93#  128-116
SA.CTP   89-59#
SA.DI    89-39#  129-68
SA.ERC   89-17#
SA.ERR   89-7#    98-56   130-22
SA.GO    89-84#  128-103
SA.INE   89-76#
SA.INT   89 24#  148-23
SA.LFC   89-85#
```

Cross reference table (CREF V05.01)

```
SA.MCV   89-920  128-111
SA.MP    89-400  129-69
SA MS1   89-300  149-10
SA.MSE   89-550
SA.MSG   89 250
SA.NV    89-370
SA.NVE   89-770
SA.PRG   89-480
SA.QB    89-380  129-69
SA.S1    89-110  129-44   129-68   145-10   146-11   148-26
SA.S2    89-100  129-78   145-14
SA.S3    89-90   129-86
SA.S4    89-80   129-56
SA.SM    89-410  129-69
SA.STE   89-580
SA.STP   89-280  129-19   145-13   146-14   146-15   148-23   149-10
SA.TST   89-670  129-23
SA.VCE   89-750
SA.VEC   89-230
SA.WRP   89-270  146-14   146-15
SAMVEC  137-206  138-360
SETOO   127-180  127-23
SETO1   127-19   127-210
SETO2   127-290  127-32
SETTO   102-146  125-16   127-140  130-26   134-11   135-18   137-117  137-144  148-33
SFPTBL   87-110  105-33   108-48
SM.IW   155-260
SM.LOG  155-250
SM.MAN  105-33   155-230  155-32
SM.SSF  108-48   155-240
SND.S1   96-900  129-20*  129-75   129-84
SND.S2   96-920  129-21*  129-22*
SND.S3   96-940  129-23*
SNDCMD  102-142  120-78   121-23   123-130
SO.BIT  105-33   108-48   155-180  155-32   155-32   155-32
SSTEP4   96-970  128-83*  128-109  128-125* 128-131* 148-72
ST.ABO   93-830
ST.AOL   93-940
ST.AVL   93-850
ST.CMD   93-820
ST.CMP   93-880
ST.CNT   93-910
ST.DAT   93-890
ST.DIA   93 930
ST.DRV   93-920
ST.HST   93-900
ST.MFE   93-860
ST.MSK   93-790  102-87   120-81   121-26
ST.OFL   93-840
ST.SUB   93-800
ST.SUC   93-810
ST.WPR   93-870
STIME    96-700  102-53   102-56   135-16   137-115  137-142
SVCGBL   84-1530 84-1620  84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201
         84-201  84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84 201   84-201   84-201   84-201   84-201
         84-201  84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84 201   84-201   84-201   84-201   84 201   84-201
```

Cross reference table (CREF V05.01)

```
                84-201  84-201  84-201  84-201  84-201  84 201  84-201  84-201  84-201  84 201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84 201  84-201  84-201  84-201  85-10   85-10   85-10   86-12   86-12   86-12   86-12   86-12   86-12   87-11
                87-11   87-11   87-11   87-11   87-11   96-8    96-8    96-8    97-13   97-13   97-13   97-19   97-19   97-19
                98-10   98-10   98-10   98-15   98-15   98-15   98-19   98-19   98-19   98-23   98-23   98-23   98-27   98-27
                98-27   98-32   98-32   98-32   98-37   98-37   98-37   98-42   98-42   98-42   98-55   98-55   98-55   98-60
                98-60   98-60   98-75   98-75   98-75   98-79   98-79   98-79   98-83   98-83   98-83   98-87   98-87   98-87
                98-91   98-91   98-91   98-95   98-95   98-95   98-100  98-100  98-100  98-104  98-104  98-104  98-108  98-108
                98-108  98-113  98-113  98-113  98-117  98-117  98-117  98-122  98-122  98-122  98-126  98-126  98-126  98-131
                98-131  98-131  98-165  98-165  98-165  126-11  126-11  126-11  126-32  126-32  126-32  126-42  126-42  126-42
               126-50  126-50  126-50  135-10  135-10  135-10  136-9   136-9   136-9   137-44  137-44  137-44  139-10  139-10
               139-10  140-8   140-8   140-8   141-8   141-8   141-8   142-9   142-9   142-9   154-11  154-11  154-11  155-12
               155-12  155-12  156-17  156-17  156-17  156-17#
SVCINS         84-153# 84-159# 84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201
                84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  84-201  85-10   85-10
                85-10   85-10   85-10   85-10   85-10   85-10   86-12   86-12   87-11   87-11   97-13   97-13   97-13   97-13
                97-19   97-19   97-19   97-19   98-12   98-12   98-17   98-17   98-21   98-21   98-25   98-25   98-29   98-29
                98-34   98-34   98-39   98-39   98-53   98-53   98-58   98-58   98-73   98-73   98-77   98-77   98-81   98-81
                98-85   98-85   98-89   98-89   98-93   98-93   98-97   98-97   98-102  98-102  98-106  98-106  98-111  98 111
                98-115  98-115  98-120  98-120  98-124  98-124  98-128  98-128  98-133  98-133  98-186  98-186  99-9    99-9
                99-9    99-9    99-9    99-9    99-9    99-9    99-11   99-11   99-78   99-78   99-78   99-78   99-78   99-78
                99-78   99-78   99-78   99-78   99-78   99-78   99-80   99-80   99-80   99-80   99-80   99-80   99-80   99-80
                99-80   99-80   99-80   99-80   99-82   99-82   99-82   99-82   99-82   99-82   99-82   99-82   99-82   99-82
                99-82   99-82   99-84   99-84   99-84   99-84   99-84   99-84   99-84   99-84   99-84   99-84   99-84   99-84
               102-26  102-26  102-26  102-26  102-26  102-26  102-26  102-26  102-44  102-44  102-44  102-44  102-44  102-44
               102-44  102-44  102-47  102-47  102-59  102-59  102-94  102-94  102-94  102-94  102-94  102-94  102-94  102-94
               102-112 102-112 102-112 102-112 102-112 102-112 102-112 102-112 105-36  105-36  105-37  105-37  105-66  105-66
               105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66  105-66
               108-40  108-40  112-25  112-25  112-25  112-25  112-25  112-25  112-25  112-25  119-26  119-26  119-41  119-41
               120-20  120-20  120-20  120-20  120-20  120-20  120-20  120-20  120-20  120-20  120-20  120-20  120-91  120-91
               120-91  120-91  120-91  120-91  120-91  120-91  121-35  121-35  121-35  121-35  121-35  121-35  121-35  121-35
               125-27  125-27  125-35  125-35  125-35  125-35  125-35  125-35  125-35  125-35  125-44  125-44  125-44  125-44
               125-44  125-44  125-44  125-44  126-13  126-13  126-35  126-35  126-44  126-44  126-54  126-54  128-76  128-76
               128-76  128-76  128-76  128-76  128-76  128-76  128-96  128-96  128-96  128-96  128-96  128-96  128-96  128-96
               128-122 128-122 128-122 128-122 128-122 128-122 128-122 128-122 128-128 128-128 128-128 128-128 128-128 128-128
               128-128 128-128 128-134 128-134 128-134 128-134 128-134 128-134 128-134 128-134 129-17  129-17  129-30  129-30
               129-30  129-30  129-30  129-30  129-30  129-30  129-30  129-30  129-30  129-30  129-33  129-33  129-33  129-33
               129-36  129-36  129-36  129-36  129-36  129-36  129-36  129-36  129-93  129-93  129-93  129-93  129 93  129-93
               129-93  129-93  130-30  130-30  130-40  130-40  130-40  130-40  130-40  130-40  130-40  130-40  130-51  130-51
               130-51  130-51  130-51  130-51  130-51  130-51  132-14  132-14  132-14  132-14  132-14  132-14  132-14  132-14
               132-14  132-14  132-14  132-14  134-14  134-14  134-22  134-22  134-22  134-22  134-22  134-22  134-22  134-22
               135-93  135-93  135-93  135-93  135-107 135-107 137-46  137-46  137-46  137-46  137-48  137-48  137-52  137-52
               137-52  137-52  137-54  137-54  137-58  137-58  137-58  137-58  137-60  137-60  137-65  137-65  137-65  137-65
               137-67  137-67  137-93  137-93  137-93  137-93  137-95  137-95  137-103 137-103 137-103 137-103 137-103 137-103
               137-103 137-103 137-105 137-105 137-127 137-127 137-127 137-127 137-128 137-128 137-129 137-129 137-129 137-129
               137-130 137-130 137-140 137-140 137-140 137-140 137-140 137-140 137-140 137 140 137-140 137-140 137-140 137-140
               137-146 137-146 137-146 137-146 137-184 137-184 137-184 137-184 137-185 137-185 137-351 137-351 137-351 137-351
               137-353 137-353 137-353 137-353 138-9   138-9   138-9   138-9   138-9   138-9   138-9   138-9   138-11  138-11
```

Cross reference table (CREF V05.01)

```
        138-19  138-19  138-19  138-19  138-19  138-19  138-19  138-19  138-21  138-21  138-28  138-28  138-28  138-28
        138-28  138-28  138-28  138-28  138-30  138-30  138-37  138-37  138-37  138-37  138-37  138-37  138-37  138-37
        138-39  138-39  138-42  138-42  139-12  139-12  140-14  140-14  140-14  140-14  140-16  140-16  141-10  141-10
        141-10  141-10  141-12  141-12  142-12  142-12  142-12  142-12  142-14  142-14  143-26  143-26  143-26  143-26
        144-6   144-6   144-8   144-8   144-8   144-8   144-8   144-8   144-8   144-8   144-8   144-8   144-8   144-8
        144-12  144-12  144-12  144-12  144-15  144-15  144-15  144-15  144-15  144-15  144-15  144-15  144-16  144-16
        144-19  144-19  145-8   145-8   145-17  145-17  146-7   146-7   146-29  146-29  146-29  146-29  146-29  146-29
        146-29  146-29  146-34  146-34  147-8   147-8   147-12  147-12  148-7   148-7   148-19  148-19  148-19  148-19
        148-19  148-19  148-19  148-19  148-19  148-19  148-19  148-19  148-20  148-20  148-20  148-20  148-36  148-36
        148-48  148-48  148-48  148-48  148-61  148-61  148-61  148-61  148-67  148-67  148-67  148-67  148-67  148-67
        148-67  148-67  148-70  148-70  148-70  148-70  148-77  148-77  148-77  148-77  148-77  148-77  148-77  148-77
        148-80  148-80  148-80  148-80  148-85  148-85  148-85  148-85  148-86  148-86  149-9   149-9   149-15  149-15
        150-7   150-7   150-18  150-18  151-7   151-7   151-14  151-14  151-14  151-14  151-14  151-14  151-14  151-14
        151-14  151-14  151-14  151-14  151-32  151-32  151-94  151-94  151-94  151-94  151-94  151-94  151-94  151-94
        151-128 151-128 152-23  152-23  153-17  153-17  154-11  154-11  154-27  154-27  154-27  154-27  154-27  154-27
        154-27  154-27  154-30  154-30  154-30  154-30  154-30  154-30  154-30  154-30  154-32  154-32  154-32  154-32
        154-32  154-32  154-32  154-32  154-32  154-32  154-35  154-35  154-35  154-35  154-35  154-35  154-35  154-35
        154-35  154-35  154-41  154-41  155-12  155-12  155-32  155-32  155-32  155-32  155-32  155-32  155-55  155-55
        156-17  156-17  156-17  156-17  156-17  156-17  157-14  157-14  157-14  157-14
SVCSUB  84-153#  84-161#  144-6   144-6   144-6   145-8   145-8   145-8   146-7   146-7   146-7   147-8   147-8   147-8
        148-7   148-7   148-7   149-9   149-9   149-9   150-7   150-7   150-7   151-7   151-7   151-7
SVCTAG  84-153#  84-163#  86-23   86-23   86-23   87-21   87-21   87-21   98-12   98-12   98-12   98-17   98-17   98-17
        98-21   98-21   98-21   98-25   98-25   98-25   98-29   98-29   98-29   98-34   98-34   98-34   98-39   98-39
        98-39   98-53   98-53   98-53   98-58   98-58   98-58   98-73   98-73   98-73   98-77   98-77   98-77   98-81
        98-81   98-81   98-85   98-85   98-85   98-89   98-89   98-89   98-93   98-93   98-93   98-97   98-97   98-97
        98-102  98-102  98-102  98-106  98-106  98-106  98-111  98-111  98-111  98-115  98-115  98-115  98-120  98-120
        98-120  98-124  98-124  98-124  98-128  98-128  98-128  98-133  98-133  98-133  98-186  98-186  98-186  105-66
        105-66  105-66  126-13  126-13  126-13  126-35  126-35  126-35  126-44  126-44  126-44  126-54  126-54  126-54
        135-107 135-107 135-107 138-42  138-42  138-42  139-12  139-12  139-12  140-16  140-16  140-16  141-12  141-12
        141-12  142-14  142-14  142-14  144-19  144-19  144-19  145-17  145-17  145-17  146-34  146-34  146-34  147-12
        147-12  147-12  148-86  148-86  148-86  149-15  149-15  149-15  150-18  150-18  150-18  151-32  151-32  151-32
        151-128 151-128 151-128 152-23  152-23  152-23  153-17  153-17  153-17  154-41  154-41  154-41  155-55  155-55
        155-55  157-14  157-14  157-14  157-24  157-24  157-24
SVCTST  84-153#  84-160#  143-10  143-10  143-10  152-3   152-3   152-3   153-3   153-3   153-3
T$$AU   142-9#   142-12  142-14
T$$AUT  139-10#  139-12
T$$CLE  140-8#   140-14  140-16
T$$DAT  157-14   157-14#  157-24
T$$DU   141-8#   141-10  141-12
T$$HAR  154-11   154-11#  154-41
T$$HW    86-12    86-12#   86-23
T$$INI  137-44#  137-353  138-42
T$$MSG   98-10#   98-12   98-15#  98-17   98-19#  98-21   98-23#  98-25   98-27#  98-29   98-32#  98-34   98-37#  98-39
         98-42#   98-53   98-55#  98-58   98-60#  98-73   98-75#  98-77   98-79#  98-81   98-83#  98-85   98-87#  98-89
         98-91#   98-93   98-95#  98-97   98-100#  98-102  98-104#  98-106  98-108#  98-111  98-113#  98-115  98-117#  98-120
         98-122#  98-124  98-126#  98-128  98-131#  98-133  98-165#  98-186
T$$PC   157-13#  157-25
T$$PRO  136-9#
T$$PTA  157-13#  157-14  157-14#
T$$RPT  135-10#  135-93  135-107
T$$SOF  155-12   155-12#  155-55
T$$SRV  126-11#  126-13  126-32#  126-35  126-42#  126-44  126-50#  126-54
T$$SUB  144-6#   144-19  145-8#   145-17  146-7#   146-34  147-8#   147-12  148-7#   148-86  149-9#   149-15  150-7#   150-18
        151-7#   151-32
T$$SW    87-11    87-11#   87-21
T$$TES  143-10#  143-26  151-128  152-3#  152-23  153-3#  153-17
```

Cross reference table (CREF V05.01)

T$ARGC	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	99-78	99-78	99-78	99-78#	99-78#	99-80	99-80	99-80	99-80#	99-80#	99-82
	99-82	99-82	99-82#	99-82#	99-84	99-84	99-84	99-84#	99-84#					
T$CODE	105-66	105-66	105-66	105-66#	105-66#	105-66#	154-27	154-27	154-27	154-27#	154-27#	154-27#	154-30	154-30
	154-30	154-30#	154-30#	154-30#	154-32	154-32	154-32	154-32#	154-32#	154-32#	154-35	154-35	154-35	154-35#
	154-35#	154-35#	155-32	155-32	155-32	155-32#	155-32#	155-32#						
T$ERRN	84-153#	99-9	99-9#	102-26	102-26#	102-44	102-44#	102-94	102-94#	102-112	102-112#	112-25	112-25#	120-91
	120-91#	121-35	121-35#	125-35	125-35#	125-44	125-44#	128-76	128-76#	128-96	128-96#	128-122	128-122#	128-128
	128-128#	128-134	128-134#	129-36	129-36#	129-93	129-93#	130-40	130-40#	130-51	130-51#	134-22	134-22#	137-103
	137-103#	138-9	138-9#	138-19	138-19#	138-28	138-28#	138-37	138-37#	144-15	144-15#	146-29	146-29#	148-67
	148-67#	148-77	148-77#	151-94	151-94#									
T$EXCP	105-66	105-66#	154-27	154-27#	154-30	154-30#	154-32	154-32#	154-35	154-35#				
T$FLAG	135-93	135-93#	135-93#	137-353	137-353	137-353#	137-353#	140-14	140-14	140-14#	140-14#	141-10	141-10#	141-10#
	142-12	142-12#	142-12#	143-26	143-26	143-26#	143-26#							
T$FREE	156-17	157-25#												
T$GMAN	84-153#	105-66	105-66#	105-66#										
T$HILI	105-66	105-66#	154-27	154-27#	154-30	154-30#	154-32	154-32#	154-35	154-35#				
T$LAST	84-153#	156-17#	157-13											
T$LOLI	105-66	105-66#	154-27	154-27#	154-30	154-30#	154-32	154-32#	154-35	154-35#				
T$LSYM	84-153	84-153#	86-23	87-21	98-12	98-17	98-21	98-25	98-29	98-34	98-39	98-53	98-58	98-73
	98-77	98-81	98-85	98-89	98-93	98-97	98-102	98-106	98-111	98-115	98-120	98-124	98-128	98-133
	98-186	126-13	126-35	126-44	126-54	135-107	138-42	139-12	140-16	141-12	142-14	144-19	145-17	146-34
	147-12	148-86	149-15	150-18	151-32	151-128	152-23	153-17	154-41	155-55				
T$LTNO	156-17#													
T$NEST	84-153#	86-12	86-12	86-12#	86-23	86-23	86-23	86-23#	87-11	87-11	87-11#	87-21	87-21	87-21
	87-21#	98-10	98-10	98-10#	98-12	98-12	98-12	98-12#	98-15	98-15	98-15#	98-17	98-17	98-17
	98-17#	98-19	98-19	98-19#	98-21	98-21	98-21	98-21#	98-23	98-23	98-23#	98-25	98-25	98-25
	98-25#	98-27	98-27	98-27#	98-29	98-29	98-29	98-29#	98-32	98-32	98-32#	98-34	98-34	98-34
	98-34#	98-37	98-37	98-37#	98-39	98-39	98-39	98-39#	98-42	98-42	98-42#	98-53	98-53	98-53
	98-53#	98-55	98-55	98-55#	98-58	98-58	98-58	98-58#	98-60	98-60	98-60#	98-73	98-73	98-73
	98-73#	98-75	98-75	98-75#	98-77	98-77	98-77	98-77#	98-79	98-79	98-79#	98-81	98-81	98 81
	98-81#	98-83	98-83	98-83#	98-85	98-85	98-85	98-85#	98-87	98-87	98-87#	98-89	98-89	98-89
	98-89#	98-91	98-91	98-91#	98-93	98-93	98-93	98-93#	98-95	98-95	98-95#	98-97	98-97	98-97
	98-97#	98-100	98-100	98-100#	98-102	98-102	98-102	98-102#	98-104	98-104	98-104#	98-106	98-106	98-106
	98-106#	98-108	98-108	98-108#	98-111	98-111	98-111	98-111#	98-113	98-113	98-113#	98-115	98-115	98-115
	98-115#	98-117	98-117	98-117#	98-120	98-120	98-120	98-120#	98-122	98-122	98-122#	98-124	98-124	98-124
	98-124#	98-126	98-126	98-126#	98-128	98-128	98-128	98-128#	98-131	98-131	98-131#	98-133	98-133	98-133
	98-133#	98-165	98-165	98-165#	98-186	98-186	98-186	98-186#	126-11	126-11	126-11#	126-13	126-13	126-13
	126-13#	126-32	126-32	126-32#	126-35	126-35	126-35	126-35#	126-42	126-42	126-42#	126-44	126-44	126-44
	126-44#	126-50	126-50	126-50#	126-54	126-54	126-54	126-54#	135-10	135-10	135-10#	135-107	135-107	135-107
	135 107#	136-9	136-9	136-9#	136-15	136-15	136-15	136-15#	137-44	137-44	137-44#	138-42	138-42	138-42
	138-42#	139-10	139-10	139-10#	139-12	139-12	139-12	139-12#	140-8	140-8	140-8#	140-16	140 16	140-16
	140-16#	141-8	141-8	141-8#	141-12	141-12	141-12	141-12#	142-9	142-9	142-9#	142-14	142-14	142 14
	142-14#	143-10	143-10	143-10#	144-6	144-6	144-6#	144-19	144-19	144-19	144-19#	145-8	145-8	145-8#
	145-17	145-17	145-17	145-17#	146-7	146-7	146-7#	146-34	146-34	146-34	146-34#	147-8	147-8	147-8#
	147-12	147-12	147-12	147-12#	148-7	148-7	148-7#	148-86	148-86	148-86	148-86#	149-9	149-9	149-9#
	149-15	149-15	149-15	149-15#	150-7	150-7	150-7#	150-18	150-18	150-18	150-18#	151-7	151-7	151-7#
	151-32	151-32	151-32	151-32#	151-128	151-128	151-128	151-128#	152-3	152-3	152-3#	152-23	152-23	152-23
	152-23#	153-3	153-3	153-3#	153-17	153-17	153-17	153-17#	154-11	154-11	154-11#	154-41	154-41	154-41
	154-41#	155-12	155-12	155-12#	155-55	155-55	155-55	155-55#						
T$NSO	86-12#	86-23	87-11#	87-21	98-10#	98-12	98-15#	98-17	98-19#	98-21	98-23#	98-25	98-27#	98-29
	98-32#	98-34	98-37#	98-39	98-42#	98-53	98-55#	98-58	98-60#	98-73	98-75#	98-77	98-79#	98-81
	98-83#	98-85	98-87#	98-89	98-91#	98-93	98-95#	98-97	98-100#	98-102	98-104#	98-106	98-108#	98-111
	98-113#	98-115	98-117#	98-120	98-122#	98-124	98-126#	98-128	98-131#	98-133	98-165#	98-186	126-11#	126-13
	126-32#	126-35	126-42#	126-44	126-50#	126-54	135-10#	135-107	136-9#	136-15	137-44#	138-42	139-10#	139-12
	140-8#	140-16	141-8#	141-12	142-9#	142-14	143-10#	151-128	152-3#	152-23	153-3#	153-17	154-11#	154-41

Cross reference table (CREF V05.01)

	155-12#	155-55												
T$NS1	144-6#	144-19	145-8#	145-17	146-7#	146-34	147-8#	147-12	148-7#	148-86	149-9#	149-15	150-7#	150-18
	151-7#	151-32												
T$PCNT	157-13#	157-14	157-14	157-14#										
T$PTAB	157-14	157-14#												
T$PTHV	84 201	157-25#												
T$PTNU	84-153#	157-14	157-14#	157-25	157-25									
T$SAVL	84-153#													
T$SEGL	84-153#													
T$SIZE	156-17	157-25#												
T$SUBN	84-153#	143-10#	144-6	144-6	144-6#	145-8	145-8	145-8#	146-7	146-7	146-7#	147-8	147-8	147-8#
	148-7	148-7	148-7#	149-9	149-9	149-9#	150-7	150-7	150-7#	151-7	151-7	151-7#	152-3#	153-3#
T$TAGL	84-153#													
T$TAGN	84-153#	86-12	86-12	86-12#	87-11	87-11	87-11#	98-10	98-10	98-10#	98-15	98-15	98-15#	98-19
	98-19	98-19#	98-23	98-23	98-23#	98-27	98-27	98-27#	98-32	98-32	98-32#	98-37	98-37	98-37#
	98-42	98-42	98-42#	98-55	98-55	98-55#	98-60	98-60	98-60#	98-75	98-75	98-75#	98-79	98-79
	98-79#	98-83	98-83	98-83#	98-87	98-87	98-87#	98-91	98-91	98-91#	98-95	98-95	98-95#	98-100
	98-100	98-100#	98-104	98-104	98-104#	98-108	98-108	98-108#	98-113	98-113	98-113#	98-117	98-117	98-117#
	98-122	98-122	98-122#	98-126	98-126	98-126#	98-131	98-131	98-131#	98-165	98-165	98-165#	126-11	126-11
	126-11#	126-32	126-32	126-32#	126-42	126-42	126-42#	126-50	126-50	126-50#	135-10	135-10	135-10#	136-9
	136-9	136-9#	137-44	137-44	137-44#	139-10	139-10	139-10#	140-8	140-8	140-8#	141-8	141-8	141-8#
	142-9	142-9	142-9#	143-10	143-10	143-10#	144-6	144-6	144-6#	145-8	145-8	145-8#	146-7	146-7
	146-7#	147-8	147-8	147-8#	148-7	148-7	148-7#	149-9	149-9	149-9#	150-7	150-7	150-7#	151-7
	151-7	151-7#	152-3	152-3	152-3#	153-3	153-3	153-3#	154-11	154-11	154-11#	155-12	155-12	155-12#
	157-13	157-13	157-13#	157-14	157-14	157-14	157-14	157-14#	157-14#					
T$TEMP	85-10	85-10	85-10	85-10	85-10	85-10	85-10#	85-10#	85-10#	85-10#	86-23	86-23#	87-21	87-21#
	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#	98-29	98-29#	98-34	98-34#	98-39	98-39#
	98-53	98-53#	98-58	98-58#	98-73	98-73#	98-77	98-77#	98-81	98-81#	98-85	98-85#	98-89	98-89#
	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#	98-111	98-111#	98-115	98-115#	98-120	98-120#
	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98-186#	105-66	105-66	105-66	105-66#	105-66#	105-66#
	126-13	126-13#	126-35	126-35#	126-44	126-44#	126-54	126-54#	135-93	135-93#	135-107	135-107#	136-15	136-15#
	137-353	137-353#	138-42	138-42#	139-12	139-12#	140-14	140-14#	140-16	140-16#	141-10	141-10#	141-12	141-12#
	142-12	142-12#	142-14	142-14#	143-26	143-26#	144-19	144-19#	145-17	145-17#	146-34	146-34#	147-12	147-12#
	148-86	148-86#	149-15	149-15#	150-18	150-18#	151-32	151-32#	151-128	151-128#	152-23	152-23#	153-17	153-17#
	154-27	154-27	154-27	154-27#	154-27#	154-27#	154-30	154-30	154-30	154-30#	154-30#	154-30#	154-32	154-32
	154-32	154-32#	154-32#	154-32#	154-35	154-35	154-35	154-35#	154-35#	154-35#	154-41	154-41#	155-32	155-32
	155-32	155-32#	155-32#	155-32#	155-55	155-55#								
T$TEST	84-153#	143-10	143-10	143-10#	144-6	145-8	146-7	147-8	148-7	149-9	150-7	151-7	152-3	152-3
	152-3#	153-3	153-3	153-3#	156-17									
T$TSTM	84-153#	98-12	98-17	98-21	98-25	98-29	98-34	98-39	98-53	98-58	98-73	98-77	98-81	98-85
	98-89	98-93	98-97	98-102	98-106	98-111	98-115	98-120	98-124	98-128	98-133	98-186	99-9	99-11
	99-78	99-80	99-82	99-84	102-26	102-44	102-47	102-59	102-94	102-112	105-36	105-66	108-40	112-25
	119-26	119-41	120-20	120-91	121-35	125-27	125-35	125-44	128-76	128-96	128-122	128-128	128-134	129-17
	129-30	129-33	129-36	129-93	130-30	130-40	130-51	132-14	134-14	134-22	135-107	137-46	137-52	137-58
	137-65	137-93	137-103	137-105	137-127	137-129	137-140	137-146	137-184	137-351	137-353	138-9	138-11	138-19
	138-21	138-28	138-30	138-37	138-39	138-42	139-12	140-14	140-16	141-12	142-14	143-26	144-6	144-8
	144-12	144-15	144-16	144-19	145-8	145-17	146-7	146-29	146-34	147-8	147-12	148-7	148-19	148-20
	148-36	148-48	148-61	148-67	148-70	148-77	148-80	148-85	148-86	149-9	149-15	150-7	150-18	151 7
	151-14	151-32	151-94	151-128	152-23	153-17								
T$TSTS	84-153#	143-10#	152-3#	153-3#										
T1	85-10	143-10#												
T1.1	144-6#													
T1.2	145-8#													
T1.3	146-7#													
T1.4	147-8#													
T1.5	148-7#													

L2

SEQ 0231

Cross reference table (CREF V05.01)

Cross reference table (CREF V05.01)

```
UF.RPL   91-102#
UF.SCH   91 104#
UF.SCL   91 105#
UF.WBN   91-106#
UF.WPH   91-107#
UF.WPS   91-108#
URNING   96-29#  101-17*  101-31*  101-38   102-71*
URUN     96-28#  101-16*  101-22   102-11
UTOT1   107-21#  107-35
UTOT1A  107-24   107-34#
UTOT2   107-22   107-36#
WAITMS  120-79   121-24   125-12#
WCHNG   134-8#   146-16   146-24
WCHNGD   96-100# 134-12   146-14*  146-22*
X$ALWA   84-153#
X$FALS   84-153#
X$OFFS   84-153#
X$TRUE   84-153#
X1       97-90#   98-11
X14      97-100#  98-38
X1A      97-85#   98-11
X2       97-92#   98-16
X22      97-106#  98-57
X23A     97-108#  98-61
X23B     97-113#  98-65
X24      97-114#  98-76
X25      97-116#  98-80
X26      97-118#  98-84
X27      97-121#  98-88
X28      97-123#  98-92
X29      97-124#  98-96
X2A      97-86#   98-16
X3       97-93#   98-20
X31      97-127#  98-105
X32      97-129#  98-109
X35      97-130#  98-118
X36      97-131#  98-123
X38      97-133#  98-127
X39      97-138#  98-132
X3A      97-87#   98-20
X4       97-94#   98-24
X6       97-96#   98-28
X8       97-98#   98-33
X8A      97-88#   98-33
XFCE     97-146#  98-43    98-101
XFRU     97-154#  98-51    98-72    111-106
XFRUA    97-155#  98-48
XMSG1    97-147#  98-155
XMSG2    97-148#  98-159
XPKT1    97-149#  98-141
XPKT2    97-152#  98-148
XSA      97-153# 111-118
```

Cross reference table (CREF V05.01)

```
ASSUME  84-40#   98-173  101-28  107-25  137-190 137-191 143-22  146-9   148-9   148 18
BCOMPL   1-15#   84-153# 105-37  137-128 137-130
BERROR   1-19#   84-153#
BGNAU    1-23#   84 153# 142-9
BGNAUT   1 31#   84-153# 139-10
BGNCLN   1-39#   84-153# 140-8
BGNDU    1-47#   84-153# 141-8
BGNHRD   1-55#   84-153# 154-11
BGNHW    1-66#   84-153#  86-12
BGNINI   1-77#   84-153# 137-44
BGNMOD   1-85#   84-153#
BGNMSG   1-98#   84-153#  98-10  98-15  98-19  98-23  98-27  98-32  98-37  98-42  98-55  98-60  98-75  98-79
        98-83    98-87    98-91  98-95  98-100 98-104 98-108 98-113 98-117 98-122 98-126 98-131 98-165
BGNPRO   1-106#  84-153# 136-9
BGNPTA   1-114#  84-153# 157-14
BGNRPT   1-144#  84-153# 135-10
BGNSEG   1-152#  84-153#
BGNSET   1-161#  84-153# 157-13
BGNSFT   1-182#  84-153# 155-12
BGNSRV   1-193#  84-153# 126-11  126-32  126-42  126-50
BGNSUB   1-201#  84-153# 144-6   145-8   146-7   147-8   148-7   149-9   150-7   151-7
BGNSW    1-225#  84-153#  87-11
BGNTST   1-236#  84-153# 143-10  152-3   153-3
BNCOMP   1-266#  84-153# 137-48  137-54  137-60  137-67  137-95  137-185
BNERRO   1-270#  84-153#
BREAK    1-274#  84-153# 102-47  125-27  129-17  130-30  134-14  148-36
BRESET   1-278#  84-153#
CKLOOP   1-282#  84-153# 144-16
CLOCK    1-286#  84-153# 137-127 137-129
CLOSE    1-292#  84-153#
CLRVEC   1-296#  84-153# 129-33  144-12  148-80  148-85
COMMEN   1-301#  84-153#
DELAY    1-322#  84-153#
DESCRI   1-317#  84-153#  97-19
DEVTYP   1-341#  84-153#  97-13
DISPAT   1-346#  84-153#  85-10
DISPLA   1-360#  84-153#
DOCLN    1-376#  84-153#  99-11  137-105 138-11  138-21  138-30  138-39
DODU     1-380#  84-153#
DORPT    1-385#  84-153# 102-59  119-26
END     84-80#   94-45    95-78  154-25  155-27
ENDAU    1-389#  84-153# 142-14
ENDAUT   1-401#  84-153# 139-12
ENDCLN   1-413#  84-153# 140-16
ENDCOM   1-425#  84-153#
ENDDU    1-441#  84-153# 141-12
ENDHRD   1-453#  84-153# 154-41
ENDHW    1-465#  84-153#  86-23
ENDINI   1-475#  84-153# 138-42
ENDMOD   1-487#  84-153#
ENDMSG   1-500#  84-153#  98-12  98-17  98-21  98-25  98-29  98-34  98 39  98-53  98-58  98-73  98-77  98-81
        98-85    98-89    98-93  98-97  98-102 98-106 98-111 98-115 98-120 98-124 98-128 98-133 98 186
ENDPRO   1-512#  84-153# 136-15
ENDPTA   1-520#  84-153# 157-24
ENDRPT   1-529#  84-153# 135-107
ENDSEG   1-541#  84-153#
```

Cross reference table (CREF V05.01)

```
ENDSET    1-5550  84-1530# 157-25
ENDSFT    1-5680  84-1530# 155-55
ENDSRV    1-5800  84-1530# 126-13  126-35  126-44  126-54
ENDSUB    1-5960  84-1530# 144-19  145-17  146 34  147-12  148-86  149-15  150-18  151-32
ENDSW     1 6140  84-1530#  87-21
ENDTST    1 6240  84-1530# 151-128  152-23  153-17
EQUALS    1-6420  84 1530#  88-8
ERRDF     1-7140  84-1530# 102-26  102-44  102-94  102-112  112-25  120-91  121-35  125-35  125-44  128-76  128-96  128-122
        129-36   129-93   130-40  130-51  134-22  144-15  146-29  148-67  148-77  151-94
ERRHRD    1-7180  84-1530#
ERROR     1-7220  84-1530# 119-41
ERRSF     1-7260  84-1530#  99-9  128-128  128-134  137-103  138-9  138-19  138-28  138 37
ERRSOF    1-7300  84-1530#
ERRTBL    1-7340  84-1530#  96-8
ESCAPE    1-7440  84-1530#
EXIT      1-7710  84-1530# 135-93  137-353  140-14  141-10  142-12  143-26
FEQUAL    1-8100  84-1530#
GETBYT    1-8240  84-1530#
GETPRI    1-8340  84-1530#
GETWOR    1-8290  84-1530#
GMANIA    1-8390  84-1530#
GMANID    1-8480  84-1530# 105-66
GMANIL    1-8590  84-1530#
GPHARD    1-8680  84-1530# 137-93  137-184
GPRMA     1-8740  84-1530# 154-27  154-30
GPRMD     1-9030  84-1530# 105-66  105-660  154-32  154-35
GPRML     1-9340  84-1530# 155-32
HEADER    1-9540  84-1530#  84-201
INLOOP    1 9620  84-1530#
IOSETU    1-9660  84-1530#
IOSTAR    1-9740  84-1530#
ITEM     84-750   94-30    94-31   94-32   94-33   94-34   94-35   94-36   94-37   94 38   94 39   94-40   94-41   94-42
         94-43    95-34    95-35  154-14  154-16  154-17  154-19  155-18
KT11      1-9820  84-1530#
LASTAD    1-;470  84-1530# 156-17
MSBYTE    1-D000  84-1530#  84-201   84-201   84-201   84-2010
MSCHEC    1-E180  84-1530# 135-93   135-930  137-353  137 3530  140-14  140-140  141-10  141-100  142-12  142 120  143-26  143-260
MSCNTO    1-E820  84-1530# 105-66   105-660  154-27   154-270  154-30  154-300  154-32  154-320  154-35  154-350  155-32  155-320
MSCOUN    1-D660  84-1530#  99-78    99-780   99-80    99-800   99-82   99-820   99-84   99-840
MSDATA    1-8670  84-1530#  84-201   84-201   84-201   84-201   84-201  84-201   84-201  84-201   84-201  84-201   84-201  84-201
         84-201   84-201    84-201   84-201   84-201   84-201   84-201  84-201   84-201  84-201   84-201  84-201   84-201  84-201
         84-201   84-201    84-201   84-201   84-201   84-201   84-201  84-201   84-201  84-201   84-201  84-2010  84-2010  97-13
         97-130   97-19     97-190
MSDECR    1-D290  84-1530#  86-23    86-230   87-21    87-210   98-12   98-120   98-17   98-170   98-21   98-210   98-25   98-250
         98-29    98-290    98-34    98-340   98-39    98-390   98-53   98-530   98-58   98-580   98-73   98-730   98-77   98-770
         98-81    98-810    98-85    98-850   98-89    98-890   98-93   98-930   98 97   98-970   98-102  98-1020  98-106  98-1060
         98 111   98-1110   98-115   98-1150  98-120   98-1200  98-124  98-1240  98-128  98-1280  98-133  98-1330  98-186  98-1860
        126-13   126-130   126-35   126-350  126-44   126-440  126-54  126-540  135-107  135-1070  136 15  136-150  138-42  138-420
        139-12   139-120   140-16   140-160  141-12   141-120  142-14  142-140  144-19   144-190   145-17  145-170  146-34  146-340
        147-12   147-120   148-86   148-860  149-15   149-150  150-18  150-180  151-32   151 320   151-128  151-1280  152-23  152-230
        153 17   153-170   154-41   154-410  155-55   155-550  157-14  157-140
MSDEFA    1-E700  84 1530#  105-66   105-660  154-27   154-270  154-30  154-300  154-32   154-320  154-35  154-350  155-32  155-320
MSENDE    1-D740  84-1530#  86-230   87-210   98-120   98-170   98-210  98-250   98-290   98-340   98-390  98-530   98-580  98-730
         98-770   98-810    98-850   98-890   98 930   98-970   98-1020 98-1060  98-1110  98-1150  98-1200 98-1240  98-1280 98-1330
         98 1860  126 130   126-350  126-440  126 540  135-1070 138-420 139-120  140-160  141 120  142 140  144-190  145-170 146-340
        147-120   148-860   149 150  150 180  151-320  151-1280 152-230 153-170  154 410  155 550
```

Cross reference table (CREF V05.01)

```
M%ERRI    1-249#  84-153#  99-9     99-9#   102-26  102 26#  102-44  102-44#  102 94  102 94#  102-112  102-112#  112-25  112 25#
        120 91    120-91#  121-35   121-35#  125-35  125-35#  125-44  125-44#  128-76  128-76#  128 96   128 96#   128-122 128-122#
        128 128   128-128#  128-134  128-134#  129-36  129-36#  129-93  129-93#  130-40  130-40#  130-51   130-51#   134-22  134-22#
        137-103   137-103#  138-9    138-9#   138-19  138-19#  138 28  138-28#  138-37  138-37#  144-15   144 15#   146-29  146-29#
        148 67    148-67#  148-77   148-77#  151-94  151 94#
M%ESCA    1-D06#  84-153#
M%ESCS    1-D10#  84-153#
M%EXCP    1-E01#  84-153#  105-66   105-66   105-66#  154-27  154-27  154-27#  154-30  154-30  154-30#  154-32  154-32  154-32#
        154-35    154-35   154-35#
M%EXIT    1-D14#  84-153#  135-93#  137-353  137-353#  140-14  140-14#  141-10#  142-12#  143-26#  143-26#
M%EXSE    1-D22#  84-153#  135-93#  137-353#  140-14#  141-10#  142-12#  143-26#
M%EXTJ    1-D18#  84-153#  135-93   135-93#  137-353#  140-14#  141-10  141-10#  142-12  142-12#  143-26#
M%GEN     1-D38#  84-153#  84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84 201   84-201   84-201   84-201
        84-201    84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201
        84-201    84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201
        84-201#   84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#
        84-201#   84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  85-10    85-10#   86-12    86 12    86-12#   86-12#
        84-201#   84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  96-8     96-8#    97-13    97-13#   97-19    97-19#
        86-23     86-23#   87-11    87-11    87-11#   87-11#   87-21    87-21#   96-8     96-8#    97-13    97-13#   97-19    97-19#   98-23    98-23#
        98-10     98-10#   98-12    98-12#   98-15    98-15#   98-17    98-17#   98-19    98-19#   98-21    98-21#   98-23    98-23#   98-25    98-25#
        98-25     98-25#   98-27    98-27#   98-29    98-29#   98-32    98-32#   98-34    98-34#   98-37    98-37#   98-39    98-39#   98-42    98-42#
        98-42     98-42#   98-53    98-53#   98-55    98-55#   98-58    98-58#   98-60    98-60#   98-73    98-73#   98-75    98-75#   98-77    98-77#
        98-91     98-91#   98-93    98-93#   98-95    98-95#   98-97    98-97#   98-100   98-100#  98-102   98-102#  98-104   98-104#  98-106   98-106#
        98-122    98-122#  98-124   98-124#  98-126   98-126#  98-128   98-128#  98-131   98-131#  98-133   98-133#  98-165   98-165#  98-186   98-186#
        126-44    126-44#  126-50   126-50#  126-54   126-54#  135-10   135-10#  135-107  135-107#  136-9    136-9#   137-44   137-44#  138-42   138-42#
        142-9     142-9#   142-14   142-14#  143-10   143-10#  144-6    144-6#   144 19   144-19#   145-8    145-8#   145-17   145-17#  146-7    146-7#
        149-15    149-15#  150-7    150-7#   150-18   150-18#  151-7    151-7#   151-32   151 32#   151 128  151-128#  152-3    152-3#   152-23   152-23#
        156-17    156-17#  157-14   157-14#  157-24   157-24#
M%GENB    1-C38#  84-153#  105-66   105-66#
M%GETS    1-D35#  84-153#  86-23    86-23#   87-21    87-21#   98-12    98-12#   98-17    98-17#   98-21    98-21#   98-25    98-25#   98-29    98-29#
        98-29#    98-29#   98-34    98-34#   98-39    98-39#   98-53    98-53#   98-58    98-58#   98-73    98-73#   98-77    98-77#   98-81    98-81#
        98-81     98-81#   98-85    98-85#   98-89    98-89#   98-93    98-93#   98-97    98-97#   98-102   98-102#  98-106   98-106#  98-111   98-111#
        98-111    98-111#  98-115   98-115#  98-120   98-120#  98-124   98-124#  98-126   98-128#  98-133   98-133#  98-186   98-186#  126-13   126-13#
        126-35    126-35#  126-44   126-44#  126-54   126-54#  126-54#  135-107  135-107#  136-15   136-15#  138-42   138-42#  139-12   139-12#
        140-16    140-16#  141-12   141-12#  142-14   142-14#  142-14#  144-19   144-19#   145-17   145 17#  146-34   146-34#  147-12   147-12#
        148-86    148-86#  149-15   149-15#  150-18   150-18#  150-18#  151-32   151-32#   151-128  151-128#  152-23   152-23#  153-17   153-17#
        154-41    154-41#  155-55   155-55#
M%GETT    1 877#  84-153#  135-93#  137-353#  140-14#  141-10#  142-12#  143-26#
M%GNGB    1 C02#  84-153#  84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201
        84-201    84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201
        84-201    84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201   84-201
        84-201#   84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#
        84-201#   84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  85-10    85 10#   86-12    86-12    86-12#   87-11
        84-201#   84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  84-201#  98-10    98-10#   98-15    98-15#   98-19    98-19#
        87-11     87-11#   96-8     96-8#    97-13    97-13#   97-19    97-19#   97-19#   98-10    98-10#   98-15    98-15#   98-19    98-19#   98-60    98-60#
        98-23     98-23#   98-27    98-27#   98-32    98-32#   98-37    98-37#   98-37#   98-42    98-42#   98-55    98-55#   98-60    98-60#   98-100   98-100#
        98-75     98-75#   98-79    98-79#   98-83    98-83#   98-87    98-87#   98-87#   98-91    98-91#   98-95    98-95#   98-100   98-100#  98-131   98-131#
        98 104    98-104#  98 108   98-108#  98-113   98-113#  98-117   98-117#  98-117#  98-122   98 122#  98 126   98-126#  98-131   98-131#  136-9    136-9#
        98-165    98-165#  126 11   126-11#  126-32   126-32#  126-42   126-42#  126-42#  126-50   126 50#  135-10   135-10#  136-9    136-9#   155-12   155-12#
        137-44    137 44#  139 10   139-10#  140 8    140-8#   141-8    141-8#   141-8#   142 9    142 9#   154 11   154-11#  155-12   155-12#
```

Cross reference table (CREF V05.01)

	156-17	156-17#												
M$GNIN	: D49#	84-153#	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84 201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201	84-201
	84 201	84-201	84-201	84-201	84-201	84-201	84-201	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84 201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#
	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	84-201#	85-10	85-10	85-10	85-10	85-10#	85-10#
	85-10#	85-10#	86-12	86-12#	87-11	87-11#	97-13	97-13	97-13#	97-13#	97-19	97-19	97-19#	97-19#
	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#	98-29	98-29#	98-34	98-34#	98-39	98-39#
	98-53	98-53#	98-58	98-58#	98-73	98-73#	98-77	98-77#	98-81	98-81#	98-85	98-85#	98-89	98-89#
	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#	98-111	98-111#	98-115	98-115#	98-120	98-120#
	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98-186#	99-9	99-9	99-9	99-9	99-9#	99-9#
	99-9#	99-9#	99-9#	99-11	99-11#	99-78	99-78	99-78	99-78	99-78	99-78	99-78#	99-78#	99-78#
	99-78#	99-78#	99-80	99-80	99-80	99-80	99-80	99-80#	99-80#	99-80#	99-80#	99-80#	99-80#	99-82
	99-82	99-82	99-82	99-82	99-82	99-82#	99-82#	99-82#	99-82#	99-82#	99-84	99-84	99-84	99-84
	99-84	99-84	99-84#	99-84#	99-84#	99-84#	99-84#	102-26	102-26	102-26	102-26	102-26#	102-26#	102-26#
	102-26#	102-26#	102-44	102-44	102-44	102-44	102-44#	102-44#	102-44#	102-44#	102-44#	102-47	102-47#	102-59
	102-59#	102-94	102-94	102-94	102-94	102-94#	102-94#	102-94#	102-94#	102-94#	102-112	102-112	102-112	102-112
	102-112#	102-112#	102-112#	102-112#	102-112#	105-36	105-36#	105-37	105-37#	105-66	105-66	105-66	105-66	105-66
	105-66	105-66	105-66	105-66#	105-66#	105-66#	105-66#	108-40	108-40#	108-40#	112-25	112-25	112-25	112-25
	112-25#	112-25#	112-25#	112-25#	112-25#	119-26	119-26#	119-41	119-41#	120-20	120-20	120-20	120-20	120-20
	120-20	120-20#	120-20#	120-20#	120-20#	120-20#	120-20#	120-91	120-91	120-91	120-91	120-91#	120-91#	120-91#
	120-91#	120-91#	121-35	121-35	121-35	121-35	121-35#	121-35#	121-35#	125-27	125-27#	125-35		
	125-35	125-35	125-35	125-35#	125-35#	125-35#	125-35#	125-35#	125-44	125-44	125-44	125-44	125-44#	125-44#
	125-44#	125-44#	125-44#	126-13	126-13#	126-35	126-35#	126-44	126-44#	126-54	126-54#	128-76	128-76	128-76
	128-76	128-76#	128-76#	128-76#	128-76#	128-76#	128-96	128-96	128-96	128-96	128-96#	128-96#	128-96#	128-96#
	128-96#	128-122	128-122	128-122	128-122	128-122#	128-122#	128-122#	128-122#	128-128	128-128	128-128	128-128	
	128-128#	128-128#	128-128#	128-128#	128-128#	128-134	128-134	128-134	128-134	128-134#	128-134#	128-134#	128-134#	128-134#
	129-17	129-17#	129-30	129-30	129-30	129-30	129-30	129-30	129-30#	129-30#	129-30#	129-30#	129-30#	129-30#
	129-33	129-33	129-33#	129-33#	129-36	129-36	129-36	129-36	129-36#	129-36#	129-36#	129-36#	129-36#	129-93
	129-93	129-93	129-93	129-93#	129-93#	129-93#	129-93#	129-93#	130-30	130-30#	130-40	130-40	130-40	130-40
	130-40#	130-40#	130-40#	130-40#	130-40#	130-51	130-51	130-51	130-51	130-51#	130-51#	130-51#	130-51#	130-51#
	132-14	132-14	132-14	132-14	132-14	132-14	132-14#	132-14#	132-14#	132-14#	132-14#	132-14#	134-14	134-14#
	134-22	134-22	134-22	134-22	134-22#	134-22#	134-22#	134-22#	134-22#	135-93	135-93	135-93#	135-93#	135-107
	135-107#	137-46	137-46	137-46#	137-46#	137-48	137-48#	137-52	137-52	137-52#	137-52#	137-54	137-54#	137-58
	137-58	137-58#	137-58#	137-60	137-60#	137-65	137-65	137-65#	137-65#	137-67	137-67#	137-93	137-93	137-93#
	137-93#	137-93#	137-95	137-95#	137-103	137-103	137-103	137-103	137-103#	137-103#	137-103#	137-103#	137-103#	137-105
	137-105#	137-127	137-127	137-127#	137-127#	137-127#	137-128	137-128#	137-129	137-129	137-129#	137-129#	137-129#	137-130
	137-130#	137-140	137-140	137-140	137-140	137-140	137-140	137-140#	137-140#	137-140#	137-140#	137-140#	137-140#	137 146
	137-146	137-146#	137-146#	137-184	137-184	137-184#	137-184#	137-184#	137-185	137-185#	137-351	137-351	137-351#	137-351#
	137-353	137-353	137-353#	137-353#	138-9	138-9	138-9	138-9	138-9#	138-9#	138-9#	138-9#	138-9#	138-11
	138-11#	138-19	138-19	138-19	138-19	138-19#	138-19#	138-19#	138-19#	138-19#	138-21	138-21#	138-28	138-28
	138-28	138-28	138-28#	138-28#	138-28#	138-28#	138-28#	138-30	138-30#	138-37	138-37	138-37	138-37	138-37#
	138-37#	138-37#	138-37#	138-37#	138-39	138-39#	138-42	138-42#	139-12	139-12#	140-14	140-14	140-14#	140-14#
	140-16	140-16#	141-10	141-10	141-10#	141-10#	141-12	141-12#	142-12	142-12	142-12#	142-12#	142-14	142-14#
	143-26	143-26	143-26#	143-26#	144-6	144-6#	144-8	144-8	144-8	144-8	144-8	144-8	144-8#	144-8#
	144-8#	144-8#	144-8#	144-8#	144-12	144-12	144-12#	144-12#	144-15	144-15	144-15	144-15	144-15#	144-15#
	144-15#	144-15#	144-15#	144-16	144-16#	144-19	144-19#	145-8	145-8#	145 17	145-17#	146-7	146-7#	146-29
	146-29	146-29	146-29	146-29#	146-29#	146-29#	146-29#	146-29#	146-34	146-34#	147-8	147-8#	147-12	147-12#
	148-7	148-7#	148-19	148-19	148-19	148-19	148-19	148-19	148-19#	148-19#	148-19#	148-19#	148-19#	148-19#
	148-20	148-20	148-20#	148-20#	148-36	148-36#	148-48	148-48	148-48#	148-48#	148-61	148-61	148-61#	148-61#
	148-67	148-67	148-67	148-67	148-67#	148-67#	148-67#	148-67#	148-67#	148-70	148-70	148-70#	148-70#	148-77
	148-77	148-77	148-77	148-77#	148-77#	148-77#	148-77#	148-77#	148-80	148-80	148-80#	148-80#	148-85	148-85
	148-85#	148-85#	148 86	148-86#	149-9	149-9#	149-15	149-15#	150-7	150 7#	150 18	150-18#	151-7	151-7#
	151-14	151-14	151 14	151-14	151-14	151-14	151-14#	151-14#	151-14#	151-14#	151-14#	151 14#	151-32	151-32#

	151-94	151-94	151-94	151-94	151-94#	151-94#	151-94#	151-94#	151-94#	151-128	151-128#	152-23	152-23#	153-17
	153-17#	154-11	154-11#	154-27	154-27	154-27	154-27	154-27#	154-30	154-30	154-30	154-30	154-30#	154-32
	154 32	154-32	154-32	154-32	154-32#	154-35	154-35	154-35	154-35	154-35	154-35#	154-41	154-41#	155-12
	155-12#	155-32	155-32	155-32	155-32#	155-55	155-55#	156-17	156-17	156-17	156-17#	157-14	157-14	157-14#
	157-14#													
M$GNLS	1 C13#	84-153#	105-66	105-66#										
M$GNSU	1-B98#	84-153#	144-6	144-6#	145-8	145-8#	146-7	146-7#	147-8	147-8#	148-7	148-7#	149-9	149-9#
	150-7	150-7#	151-7	151-7#										
M$GNTA	1-B90#	84-153#	86-23	86-23#	87-21	87-21#	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#
	98-29	98-29#	98-34	98-34#	98-39	98-39#	98-53	98-53#	98-58	98-58#	98-73	98-73#	98-77	98-77#
	98-81	98-81#	98-85	98-85#	98-89	98-89#	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#
	98-111	98-111#	98-115	98-115#	98-120	98-120#	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98-186#
	126-13	126-13#	126-35	126-35#	126-44	126-44#	126-54	126-54#	135-107	135-107#	138-42	138-42#	139-12	139-12#
	140-16	140-16#	141-12	141-12#	142-14	142-14#	144-19	144-19#	145-17	145-17#	146-34	146-34#	147-12	147-12#
	148-86	148-86#	149-15	149-15#	150-18	150-18#	151-32	151-32#	151-128	151-128#	152-23	152-23#	153-17	153-17#
	154-41	154-41#	155-55	155-55#	157-14	157-14#	157-24	157-24#						
M$GNTE	1-B94#	84-153#	143-10	143-10#	152-3	152-3#	153-3	153-3#						
M$HAPT	1-A39#	84-153#	84-201	84-201#										
M$HNAP	1-B24#	84-153#	84-201	84-201#										
M$INCR	1-D26#	84-153#	86-12	86-12	86-12#	86-12#	87-11	87-11	87-11#	87-11#	98-10	98-10	98-10#	98-10#
	98-12#	98-15	98-15	98-15#	98-15#	98-17#	98-19	98-19	98-19#	98-19#	98-21#	98-23	98-23	98-23#
	98-23#	98-25#	98-27	98-27	98-27#	98-27#	98-29#	98-32	98-32	98-32#	98-32#	98-34#	98-37	98-37
	98-37#	98-37#	98-39#	98-42	98-42	98-42#	98-42#	98-53#	98-55	98-55	98-55#	98-55#	98-53#	98-60
	98-60	98-60#	98-60#	98-73#	98-75	98-75	98-75#	98-75#	98-77#	98-79	98-79	98-79#	98-79#	98-81#
	98-83	98-83	98-83#	98-83#	98-85#	98-87	98-87	98-87#	98-87#	98-89#	98-91	98-91	98-91#	98-91#
	98-93#	98-95	98-95	98-95#	98-95#	98-97#	98-100	98-100	98-100#	98-100#	98-102#	98-104	98-104	98-104#
	98-104#	98-106#	98-108	98-108	98-108#	98-108#	98-111#	98-113	98-113	98-113#	98-113#	98-115#	98-117	98-117
	98-117#	98-117#	98-120#	98-122	98-122	98-122#	98-122#	98-124#	98-126	98-126	98-126#	98-126#	98-128#	98-131
	98-131	98-131#	98-131#	98-133#	98-165	98-165	98-165#	98-165#	98-186#	99-9#	99-11#	99-78#	99-80#	99-82#
	99-84#	102-26#	102-44#	102-47#	102-59	102-94#	102-112#	105-36#	105-66	105-66#	105-66#	108-40#	112-25#	119-26#
	119-41#	120-20#	120-91#	121-35#	125-27	125-35#	125-44#	126-11	126-11	126-11#	126-11#	126-32	126-32	126 32#
	126-32#	126-42	126-42	126-42#	126-42#	126-50	126-50	126-50#	126-50#	128-76#	128-96#	128-122#	128-128#	128-134#
	129-17#	129-30#	129-33#	129-36#	129-93#	130-30#	130-40#	130-51#	132-14#	134-14#	134-22#	135-10	135-10	135-10#
	135-10#	135-107#	136-9	136-9	136-9#	136-9#	137-44	137-44	137-44#	137-44#	137-46#	137-52#	137-58#	137-65#
	137-93#	137-103#	137-105#	137-127#	137-129#	137-140#	137-146#	137-184#	137-351#	137-353#	138-9#	138-11#	138-19#	138-21#
	138-28#	138-30#	138-37#	138-39#	138-42#	139-10	139-10	139-10#	139-10#	139-12#	140-8	140-8	140-8#	140-8#
	140-14#	140-16#	141-8	141-8	141-8#	141-8#	141-12#	142-9	142-9	142-9#	142-9#	142-14#	143-10	143-10
	143-10	143-10#	143-10#	143-10#	143-26#	144-6	144-6	144-6	144-6#	144-6#	144-6#	144-8#	144-12#	144-15#
	144-16#	144-19#	145-8	145-8	145-8	145-8#	145-8#	145-8#	145-17#	146-7	146-7	146-7	146-7#	146-7#
	146-7#	146-29#	146-34#	147-8	147-8	147-8	147-8#	147-8#	147-8#	147-12#	148-7	148-7	148-7	148-7#
	148-7#	148-7#	148-19#	148-20#	148-36#	148-48#	148-61#	148-67#	148-70#	148-77#	148-80#	148-85#	148-86#	149-9
	149-9	149-9	149-9#	149-9#	149-9#	149-15#	150-7	150-7	150-7	150-7#	150-7#	150-7#	150-18#	151-7
	151-7	151-7	151-7#	151-7#	151-7#	151-14#	151-32#	151-94#	151-128#	152-3	152-3	152-3	152-3#	152-3#
	152-3#	152-23#	153-3	153-3	153-3	153-3#	153-3#	153-3#	153-17#	154-11	154-11	154-11#	154-11#	155-12
	155-12	155-12#	155-12#	157-13	157-13#	157-14	157-14	157-14	157-14#					
M$IOSE	1-A00#	84-153#												
M$LDRO	1-C42#	84-153#	129-33	129-33#	137-46	137-46#	137-52	137-52#	137-58	137-58#	137-65	137-65#	137-93	137-93#
	137-127	137-127#	137-129	137-129#	137-184	137-184#	137-351	137-351#	144-12	144-12#	148-20	148-20#	148-48	148-48#
	148-61	148-61#	148-70	148-70#	148-80	148-80#	148-85	148-85#						
M$MASK	1-a71#	84-153#												
M$MCHI	1-40	84-153	84-153#	84-153#										
M$MCLO	1-a24#	84-153	84-153#	84-153#										
M$MSK1	1-a77#	84-153#												
M$POP	1-881#	84-153#	86-23	86-23#	87-21	87-21#	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#
	98-29	98-29#	98-34	98-34#	98-39	98-39#	98-53	98-53#	98-58	98-58#	98 73	98-73#	98-77	98-77#
	98-81	98-81#	98-85	98-85#	98-89	98-89#	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#

Cross reference table (CREF V05.01)

	98-111	98-111#	98-115	98-115#	98-120	98-120#	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98 186#
	126-13	126-13#	126-35	126-35#	126-44	126-44#	126-54	126-54#	135-107	135-107#	136-15	136-15#	138-42	138-42#
	139 12	139-12#	140-16	140-16#	141-12	141-12#	142-14	142-14#	144-19	144-19#	145-17	145-17#	146-34	146-34#
	147-12	147-12#	148-86	148-86#	149-15	149-15#	150-18	150-18#	151-32	151-32#	151-128	151-128#	152-23	152-23#
	153-17	153-17#	154-41	154-41#	155-55	155-55#								
M$PRIN	1-@36#	84-153#	99-78	99-78#	99-80	99-80#	99-82	99-82#	99-84	99-84#				
M$PUSH	1-@31#	84-153#	86-12	86-12#	87-11	87-11#	98-10	98-10#	98-15	98-15#	98-19	98-19#	98-23	98-23#
	98-27	98-27#	98-32	98-32#	98-37	98-37#	98-42	98-42#	98-55	98-55#	98-60	98-60#	98-75	98-75#
	98-79	98-79#	98-83	98-83#	98-87	98-87#	98-91	98-91#	98-95	98-95#	98-100	98-100#	98-104	98-104#
	98-108	98-108#	98-113	98-113#	98-117	98-117#	98-122	98-122#	98-126	98-126#	98-131	98-131#	98-165	98-165#
	126-11	126-11#	126-32	126-32#	126-42	126-42#	126-50	126-50#	135-10	135-10#	136-9	136-9#	137-44	137-44#
	139-10	139-10#	140-8	140-8#	141-8	141-8#	142-9	142-9#	143-10	143-10#	144-6	144-6#	145-8	145-8#
	146-7	146-7#	147-8	147-8#	148-7	148-7#	149-9	149-9#	150-7	150-7#	151-7	151-7#	152-3	152-3#
	153-3	153-3#	154-11	154-11#	155-12	155-12#								
M$PUT	1-C72#	84-153#	99-78	99-78	99-78	99-78#	99-80	99-80	99-80	99-80#	99-82	99-82	99-82	99-82#
	99-84	99-84	99-84	99-84#	120-20	120-20	120-20	120-20	120-20	129-30	129-30	129-30	129-30	129-30#
	132-14	132-14	132-14	132-14	132-14#	137-140	137-140	137-140	137-140	137-140#	144-8	144-8	144-8	144-8
	144-8#	148-19	148-19	148-19	148-19	148-19#	151-14	151-14	151-14	151-14	151-14#			
M$PUT1	1-C81#	84-153#	99-78	99-78	99-78	99-78#	99-78#	99-78#	99-80	99-80	99-80	99-80#	99-80#	99-80#
	99-82	99-82	99-82	99-82#	99-82#	99-82#	99-84	99-84	99-84	99-84#	99-84#	99-84#	120-20	120-20
	120-20	120-20	120-20#	120-20#	120-20#	120-20#	129-30	129-30	129-30	129-30	129-30#	129-30#	129-30#	129-30#
	132-14	132-14	132-14	132-14	132-14#	132-14#	132-14#	132-14#	137-140	137-140	137-140	137-140	137-140#	137-140#
	137-140#	137-140#	144-8	144-8	144-8	144-8	144-8#	144-8#	144-8#	144-8#	148-19	148-19	148-19	148-19
	148-19#	148-19#	148-19#	148-19#	151-14	151-14	151-14	151-14	151-14#	151-14#	151-14#	151-14		
M$RADI	1-D77#	84-153#	105-66	105-66#	154-27	154-27#	154-30	154-30#	154-32	154-32#	154-35	154-35#	155-32	155-32#
M$RBRO	1-C52#	84-153#												
M$RNRO	1-C62#	84-153#	108-40	108-40#	137-93	137-93#	137-127	137-127#	137-129	137-129#	137-146	137-146#	137-184	137-184#
M$SETS	1-D32#	84-153#	86-12	86-12#	87-11	87-11#	98-10	98-10#	98-15	98-15#	98-19	98-19#	98-23	98-23#
	98-27	98-27#	98-32	98-32#	98-37	98-37#	98-42	98-42#	98-55	98-55#	98-60	98-60#	98-75	98-75#
	98-79	98-79#	98-83	98-83#	98-87	98-87#	98-91	98-91#	98-95	98-95#	98-100	98-100#	98-104	98-104#
	98-108	98-108#	98-113	98-113#	98-117	98-117#	98-122	98-122#	98-126	98-126#	98-131	98-131#	98-165	98-165#
	126-11	126-11#	126-32	126-32#	126-42	126-42#	126-50	126-50#	135-10	135-10#	136-9	136-9#	137-44	137-44#
	139-10	139-10#	140-8	140-8#	141-8	141-8#	142-9	142-9#	143-10	143-10#	144-6	144-6#	145-8	145-8#
	146-7	146-7#	147-8	147-8#	148-7	148-7#	149-9	149-9#	150-7	150-7#	151-7	151-7#	152-3	152-3#
	153-3	153-3#	154-11	154-11#	155-12	155-12#								
M$STAR	1-A33#	84-153#												
M$SVC	1-C33#	84-153#	98-12	98-12#	98-17	98-17#	98-21	98-21#	98-25	98-25#	98-29	98-29#	98-34	98-34#
	98-39	98-39#	98-53	98-53#	98-58	98-58#	98-73	98-73#	98-77	98-77#	98-81	98-81#	98-85	98-85#
	98-89	98-89#	98-93	98-93#	98-97	98-97#	98-102	98-102#	98-106	98-106#	98-111	98-111#	98-115	98-115#
	98-120	98-120#	98-124	98-124#	98-128	98-128#	98-133	98-133#	98-186	98-186#	99-9	99-11	99-11#	99-78
	99-78#	99-80	99-80#	99-82	99-82#	99-84	99-84#	102-26	102-44	102-47	102-47#	102-59	102-59#	102-94
	102-112	105-36	105-36#	105-66	105-66#	108-40	108-40#	112-25	119-26	119-26#	119-41	119-41#	120-20	120-20#
	120-91	121-35	125-27	125-27#	125-35	125-44	128-76	128-96	128-122	128-128	128-134	129-17	129-17#	129-30
	129-30#	129-33	129-33#	129-36	129-93	130-30	130-30#	130-40	130-51	132-14	132-14#	134-14	134-14#	134-22
	135-93#	135-107	135-107#	137-46	137-46#	137-52	137-52#	137-58	137-58#	137-65	137-65#	137-93	137-93#	137-103
	137-105	137-105#	137-127	137-127#	137-129	137-129#	137-140	137-140#	137-146	137-146#	137-184	137-184#	137-351	137-351#
	137-353	137-353#	138-9	138-11	138-11#	138-19	138-21	138-21#	138-28	138-30	138-30#	138-37	138-39	138-39#
	138-42	138-42#	139-12	139-12#	140-14	140-14#	140-16	140-16#	141-10#	141-12	141-12#	142-12#	142-14	142-14#
	143-26	143-26#	144-6	144-6#	144-8	144-8#	144-12	144-12#	144-15	144-16	144-16#	144-19	144-19#	145-8
	145-8#	145-17	145-17#	146-7	146-7#	146-29	146-34	146-34#	147-8	147-8#	147-12	147-12#	148-7	148-7#
	148-19	148-19#	148-20	148-20#	148-36	148-36#	148-48	148-48#	148-61	148-61#	148-67	148-70	148-70#	148-77
	148-80	148-80#	148-85	148-85#	148-86	148-86#	149-9	149-9#	149-15	149-15#	150-7	150-7#	150-18	150-18#
	151-7	151-7#	151-14	151-14#	151-32	151-32#	151-94	151-128	151-128#	152-23	152-23#	153-17	153-17#	
M$TLAB	1-C29#	84-153#	98-12#	98-17#	98-21#	98-25#	98-29#	98-34#	98-39#	98-53#	98-58#	98-73#	98-77#	98-81#
	98-85#	98-89#	98-93#	98-97#	98-102#	98-106#	98-111#	98-115#	98-120#	98-124#	98-128#	98-133#	98 186#	99-9#
	99-11#	99-78#	99-80#	99-82#	99-84#	102-26#	102-44#	102-47#	102-59#	102-94#	102-112#	105 36#	105-66#	108-40#

Cross reference table (CREF V05.01)

```
                112-25#  119-26#  119-41#  120-20#  120-91#  121-35#  125-27#  125-35#  125-44#  128-76#  128-96#  128-122# 128-128# 128-134#
                129-17#  129-30#  129-33#  129-36#  129-93#  130-30#  130-40#  130-51#  132-14#  134-14#  134-22#  135-107# 137-46#  137-52#
                137-58#  137-65#  137-93#  137-103# 137-105# 137-127# 137-129# 137-140# 137-146# 137-184# 137-351# 137-353# 138-9#   138-11#
                138-19#  138-21#  138-28#  138-29#  138-33#  138-36#  138-37#  138-39#  138-42#  139-12#  140-14#  140-16#  141-12#  142-14#
                144 8#   144-12#  144-15#  144-16#  144-19#  145-8#   145-17#  146-7#   146-29#  146-34#  147-8#   147-12#  148-7#   148-19#
                148 20#  148-36#  148-48#  148-61#  148-67#  148-70#  148-77#  148-80#  148-85#  148-86#  149-9#   149-15#  150-7#   150-18#
                151-7#   151-14#  151-32#  151-94#  151-128# 152-23#  153-17#
M$TSTL            1-C21#  84-153#  98-12    98-12#   98-17    98-17#   98-21    98-21#   98-25    98-25#   98-29    98-29#   98-34    98-34#
                  98-39    98-39#   98-53    98-53#   98-58    98-58#   98-73    98-73#   98-77    98-77#   98-81    98-81#   98-85    98-85#
                  98-89    98-89#   98-93    98-93#   98-97    98-97#   98-102   98-102#  98-106   98-106#  98-111   98-111#  98-115   98-115#
                  98-120   98-120#  98-124   98-124#  98-128   98-128#  98-133   98-133#  98-186   98-186#  99-9     99-9#    99-9#    99-11
                  99-11#   99-78    99-78#   99-80    99-80#   99-82    99-82#   99-84    99-84#   102-26   102-26#  102-26#  102-44   102-44#
                102-44#  102-47   102-47#  102-59   102-59#  102-94   102-94#  102-94#  102-112  102-112# 102-112# 105-36   105-36#  105-66
                105-66#  108-40   108-40#  112-25   112-25#  112-25#  119-26   119-26#  119-41   119-41#  120-20   120-20#  120-91   120-91#
                120-91#  121-35   121-35#  121-35#  125-27   125-27#  125-35   125-35#  125-35#  125-44   125-44#  125-44#  128-76   128-76#
                128-76#  128-96   128-96#  128-96#  128-122  128-122#  128-122#  128-128  128-128#  128-128#  128-134  128-134#  128-134#  129-17
                129-17#  129-30   129-30#  129-33   129-33#  129-36   129-36#  129-36#  129-93   129-93#  129-93#  130-30   130-30#  130-40
                130-40#  130-40#  130-51   130-51#  130-51#  132-14   132-14#  134-14   134-14#  134-22   134-22#  134-22#  135-107  135-107#
                137-46   137-46#  137-52   137-52#  137-58   137-58#  137-65   137-65#  137-93   137-93#  137-103  137-103# 137-103# 137-105
                137-105# 137-127  137-127# 137-129  137-129# 137-140  137-140# 137-146  137-146# 137-184  137-184# 137-351  137-351# 137-353
                137-353# 138-9    138-9#   138-9#   138-11   138-11#  138-19   138-19#  138-19#  138 21   138-21#  138-28   138-28#  138-28#
                138-30   138-30#  138-37   138-37#  138-37#  138-39   138-39#  138-42   138-42#  139-12   139-12#  140-14   140-14#  140-16
                140-16#  141-12   141-12#  142-14   142-14#  143-26   143-26#  144-6    144-6#   144-8    144-8#   144-12   144-12#  144-15
                144-15#  144-15#  144-16   144-16#  144-19   144-19#  145-8    145-8#   145-17   145-17#  146-7    146-7#   146-29   146-29#
                146-29#  146-34   146-34#  147-8    147-8#   147-12   147-12#  148-7    148-7#   148-19   148-19#  148-20   148-20#  148-36
                148-36#  148-48   148-48#  148-61   148-61#  148-67   148-67#  148-67#  148-70   148-70#  148-77   148-77#  148-77#  148-80
                148-80#  148-85   148-85#  148-86   148-86#  149-9    149-9#   149-15   149-15#  150-7    150-7#   150-18   150-18#  151-7
                151-7#   151-14   151-14#  151-32   151-32#  151-94   151-94#  151-94#  151-128  151-128# 152-23   152-23#  153-17   153-17#
M$WORD            1-C94#  84-153#  84-201   84-201#  85-10    85-10    85-10    85-10    85-10#   99-9     99-9     99-9     99-9#    102-26
                102-26   102-26   102-26#  102-44   102-44   102-44   102-44#  102-94   102-94   102-94   102-94#  102-112  102-112  102-112
                102-112# 105-66   105-66   105-66#  105-66#  112-25   112-25   112-25   112-25#  120-91   120-91   120-91   120-91#  121-35
                121-35   121-35   121-35#  125-35   125-35   125-35   125-35#  125-44   125-44   125-44   125-44#  128-76   128-76   128-76
                128-76#  128-96   128-96   128-96   128-96#  128-122  128-122  128-122  128-122#  128-128  128-128  128-128  128-128#  128-134
                128-134  128-134  128-134#  129-36   129-36   129-36   129-36#  129-93   129-93   129-93   129-93#  130-40   130-40   130-40
                130-40#  130-51   130-51   130-51   130-51#  134-22   134-22   134-22   134-22#  135-93   135-93#  137-103  137-103  137-103
                137-103# 137-353# 138-9    138-9    138-9    138-9#   138-19   138-19   138-19   138-19#  138-28   138-28   138-28   138-28#
                138-37   138-37   138-37   138-37#  140-14#  141-10   141-10#  142-12   142-12#  143-26#  144-15   144-15   144-15   144-15#
                146-29   146-29   146-29   146-29#  148-67   148-67   148-67   148-67#  148-77   148-77   148-77   148-77#  151-94   151-94
                151-94   151-94#  154-27   154-27#  154-30   154-30#  154-32   154-32#  154-35   154-35#  155-32   155-32#  157-14   157-14#
M$XFER            1-?82#  84-153#
MANUAL            1-;62#  84-153#  105-36
MEMORY            1-;66#  84-153#  137-146
OPEN              1-;71#  84-153#
PNT             84-146#  133-25   133-29   133-34
PNT...          84-107#  98-11    98-16    98-20    98-24    98-28    98-33    98-38    98-43    98-48    98-51    98 57    98-61    98-65
                98-72    98-76    98-80    98-84    98-88    98-92    98-96    98-101   98-105   98-109   98-118   98-123   98-127   98-132
                98-141   98-148   98-155   98-159   98-172   99-156   99-236   105-38   105-47   105-113  109-26   111-68   111-81   133-25
                133-29   133-34   135-13   137-132  148-17   148-82
PNTB            84-134#  98-11    98-16    98-20    98-24    98-28    98-33    98-38    98 43    98-48    98-51    98-57    98-61    98-65
                98-72    98-76    98-80    98-84    98-88    98-92    98-96    98-101   98-105   98 109   98-118   98-123   98-127   98-132
                98-141   98-148   98-155   98-159   98-172   111-68   111-81
PNTF            84-130#  99-156   99-236   105-38   105-47   105-113  137-132
PNTS            84-142#  135-13
PNTX            84-138#  109-26   148-17   148-82
POINTE            1 ;76#  84-153#  84-192
POP             84-31#   99-35    99-76    99-110   99-237   109-28   109-39   111-15   111-35   111 41   111-96   111-108  111-120  112-26
```

Cross reference table (CREF V05.01)

```
        112-32  113-32  114-62  114-70  115-29  116-34  117-12  117-36  118-40  118-48  119-27  119-30  119-43  119-45
        120-49  121-29  121-36  122-33  122-34  124-26  125-39  125-51  126-34  127-48  128-135 128-143 129-62  130-27
        131-16  132-25  133-30  133-35  135-91  148-62  150-13  150-15  151-31  151-105 151-117 151-123 151-124 152-18
PRINT    84-96#  98-175  99-172  99-182  99-219  99-245 105-54  105-59  105-62  111-32  111-40  111-70  111-83  114-59
        114-67  117-8   117-22  117-42  133-28  133-33  133-36  135-15
PRINTB    1-<39#  84-153#  99-80
PRINTF    1-<79#  84-153#  99-78
PRINTS    1 *19#  84-153#  99-84
PRINTX    1-*59#  84-153#  99-82
PUSH     84-18#  99-30   99-69   99-104  99-108  109-21  111-9   111-24  111-26  111-93  111-105 111-117 112-14  113-14
        114-17  114-23  114-41  115-17  116-20  117-9   117-30  118-23  119-14  119-19  120-45  121-17  122-17  124-17
        125-12  127-14  128-30  129-18  130-21  131-10  132-13  133-16  133-23  135-12  148-55  150-8   151-8   151-56
        151-82  152-13
READBU    1-*99#  84-153#
READEF    1->03#  84-153#  137-46  137-52  137-58  137-65
RFLAGS    1->08#  84-153#  108-40
SETPRI    1->13#  84-153#  137-351 148-20  148-48  148-61  148-70
SETVEC    1->18#  84-153#  120-20  129-30  132-14  137-140 144-8   148-19  151-14
SLASH     1->24#  84-153#
STARS     1->38#  84-153#
SVC       1->52#  84-152#  84-153
TABLE    84-71#  94-29   95-33   154-13  155-13
XFER      1-@12#  84-153# 135-93# 137-353# 140-14# 141-10# 142-12# 143-26#
XFERF     1-@16#  84-153#
XFERT     1-@20#  84-153#
```